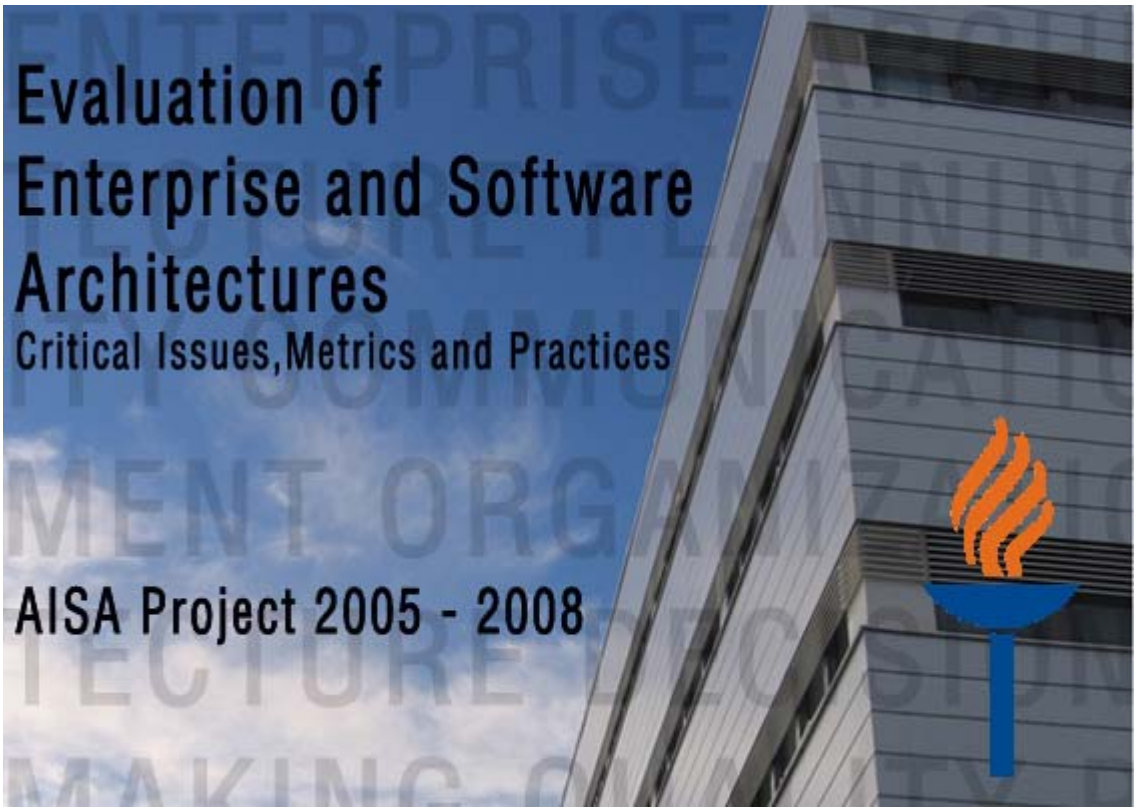


Eetu Niemi, Tanja Ylimäki & Niina Hämäinen (Eds.)
Technical Editor: Susanna Mäkinen
Publications of the Information Technology Research Institute 18, 2008.

Recommended browsers: Mozilla Firefox 1.0+ or Microsoft Internet Explorer 6.0+

Page Index

This index shows the page hierarchy of the publication.

Top of page

Author: Tanja Ylimäki

Introduction

This publication presents the AISA research project; its starting points, organization, process, and results. 'AISA' is an acronym for the project's working title: 'Assessment of Information Systems Architectures'.

The project was a joint effort by Information Technology Research Institute (ITRI) at the University of Jyväskylä and six companies - IBM Finland, Elisa, Osuuspankkikeskus (OP Bank Group Central Cooperative), A-Ware, SOK (S Group) and Tieturi. The primary aim of the project was to scrutinize the area of the quality management of both enterprise and software architectures, especially from the viewpoint of evaluation and measurement.

The project was funded by the Finnish Funding Agency for Technology and Innovation (Tekes) and the participating companies.

This publication consists of the following parts:

- Executive Summary: Summary of the project themes and results both in English and in Finnish
- AISA Project: Description of the research project
- Enterprise and Software Architecture Work: Description of the essential aspects of the architecture work both on the enterprise and software architecture level
- Quality Management in Architecture Work: Introduction to the quality issues both in general and in the architecture work context
- Evaluating Architecture Work: Description of the needs, triggers and status of architecture work evaluation
- Evaluation Practices in Architecture Work: Description of the evaluation planning components and evaluation methods incuding an evaluation model for enterprise architecture
- Metrics and Criteria for Evaluating Architecture Work: Description of the metrics and criteria for evaluating various specific targets in architecture work
- Decision-Making and Risk Management in Architecture Work: Introduction to the decision-making and risk management issues in architecture work
- Case Studies: Descriptions of the case studies conducted in three companies participating in the project
- Results: An overview of the outcome of the project including published papers and other public reports
- Lessons Learned: Conclusions of the project and suggestions for further research
- Bibliography: A variety of references on enterprise architecture, software architecture, quality management and evaluation
- Page Index: Structure of this publication

The left side click-to-expand menu can be used to navigate through the above topics. The 'Back'-button of the browser can be used to get to the previous page.

Publication Information

Publication Title: Evaluation of Enterprise and Software Architectures - Critical Issues, Metrics and Practices
Year of Publication: 2008

Editors: Eetu Niemi, Tanja Ylimäki & Niina Hämäläinen
Technical Editor: Susanna Mäkinen
Publication Series: University of Jyväskylä, Publications of the Information Technology Research Institute 18, 2008
Series Editor: Hannakaisa Isomäki
ISBN: 978-951-39-3108-7
ISSN: 1236-1615

Top of page

Executive Summary

This section gives an executive summary of the themes and results of the AISA project both in English and in Finnish. These issues are dealt with in more detail in the other sections of this publication. The actual result documents (research papers, reports and presentations) are provided in the Results section. For overall discussion, see the Lessons Learned section.

Summary in English

The AISA project, conducted by the Information Technology Research Institute (ITRI), focused on studying the quality management and evaluation aspects of both **enterprise architecture (EA)** and **software architecture (SA)**. These main research themes were clarified with the help of several sub-topics (constituting the main contents of this publication).

Before summarizing the main results, the concepts of EA and SA are briefly described.

Concepts of Enterprise and Software Architectures

Architectures are built to depict the structure of an enterprise and its components, such as processes, information systems (IS), information and IT platforms. They enable enterprises to understand and improve their complex structures, and thus more quickly and efficiently develop information systems that truly support the business, and react to changes in the business environment. EA and SA share the same fundamental idea but differ in scope.

To put it briefly, **enterprise architecture, EA, is a holistic approach for managing and developing whole enterprises** and typically describes the following four interdependent architectural dimensions:

- business architecture,
- information architecture,
- information systems architecture, and
- technology architecture.

Software Architecture, SA, on the other hand, **depicts information from all of the dimensions of EA in more details from the viewpoint of a single information system**. Thus, it has significantly narrower scope than EA. SA is a tool for planning, developing and managing software-intensive systems. It describes the components of a system, their interrelationships, external connections, and essential principles of development.

Next, the main results of the projects are introduced organized by the research topics.

Multifaceted Nature of Architecture Work

The multifaceted nature of architecture work was clarified and the following results were achieved:

- Success and failure factors for SA as well as the potential critical success factors for EA were charted. It is evident that **architecture work (including development, management, implementation and maintenance of architectures) is a vast area**, and the success and quality of both EA and SA work seem to be influenced by multiple and even interrelated factors. In both the EA and the SA domain,

  - communication,
  - commitment of both top management and other stakeholder groups (such as project managers and information system developers), and
  - architecture documentation

  seem to be perharps the most important factors.

  These and the other factors found in the study can be used as a checklist by which practitioners both in the ICT user and service provider organizations can ensure that the EA and SA development and management efforts are comprehensive, well-implemented, and have the minimum chance of failure. Additionally, the factors provide one possible starting point for the development of architecture evaluation criteria and metrics.

- A framework of the stakeholders of architecture work was developed. The framework includes EA stakeholders and their typical needs. **Architecture work is greatly influenced by the most important stakeholders** such as the top management and **responding to the needs of the stakeholders is crucial to enable the organization to benefit from EA**. In addition to money and other resources, stakeholders may also provide necessary input information for constructing architectural models. As there are a large number of potential stakeholders, the framework also divides stakeholders into three classes and discusses their common needs. In brief, stakeholder roles can be classified into

  - producers carrying out the architecture planning and development,
  - facilitators performing architecture governance and maintenance, and
  - users that utilize architecture work outputs in daily work.

The framework can be used in organizations to support **identification of the stakeholders of architecture work and their needs**. It also enables practitioners to better **manage** the variety of stakeholders.

- The possible benefits of architecture work were charted to provide support for organizations to **define the objectives and expected benefits for their architecture programs**. The most usual benefits expected to be achieved via architecture programs seem to be

    - reduced costs,
    - providing a holistic view of the enterprise,
    - improved business-IT alignment,
    - improved change management,
    - improved risk management,
    - improved interoperability and integration, and
    - shortened cycle times.

- The status of architecture work was charted in the case organizations by using the initial evaluation model for Enterprise Architecture constructed in the project. Generally, architecture work is currently under development or in initial state. It seems that

    - Organizations have defined their architectural frameworks and principles, and they provide architectural guidance to ICT development projects at least to some extent.
    - The actual architectural models are still generally under construction as well as the transition plan describing the steps required to move from the current state towards to the target state. The current state and the target state architectural models may exist on specific architecture domains or viewpoints (e.g. information architecture) but are typically inconsistent and the big picture of the organizations's EA may still be fuzzy.
    - Tool support is mainly limited to basic office tools and ICT development tools already introduced in organizations. This kind of tool set seems to be sufficient enough in the beginning of an EA program.
    - Architectural evaluation and measurement is seen as an important issue but architectural work is seldom on such a high maturity level that evaluation is considered a useful activity in the organizations. Also the lack of usable evaluation practices remains a challenge.
    - Architecture work seems to be more a project level activity than a systematic enterprise-level approach in organizations.

- The quality and quality management aspects of architecture work were studied both in the domains of EA and SA. Especially on the SA domain, a quality model for both the software architecture management process and the software architecture was constructed to support achieving high quality software architectures.

    In the EA domain, the concept of quality as well as the quality management activities are still unclear and further research is required. To put it briefly, we suggest that an **EA has high quality if it is understood, accepted and used, and the EA is measured in order to ensure that the quality requirements are met**. The existing maturity models (see e.g. GAO 2003; IAC 2005; OMB 2005) are considered as one possible means of advancing the maturity, and the quality, of architecture work.

Evaluation of Architecture Work

The evaluation of architecture work was studied in more detail and the following results were achieved:

- The possible needs and triggers for evaluation in organizations were defined. Architecture evaluations are, for example, used to

    - increase the understanding of the organization's business and ICT-environments from both financial and structural viewpoints
    - support change management, quality assurance, process planning and design, IT cost management and architectural choice making
    - manage the quality of architecture documentation, information and data structures, application and technical infrastructure, and system solutions
    - manage architecture work e.g. by conforming that the work meets expectations.

    Organization's business and ICT-related problems, questions, topics of concern and information needs are the most typical **triggers** for architecture evaluations.

- The various evaluation aspects in architecture work were clarified. Specifically, EA program evaluation seem to focus to the evaluation of

    - the benefits of EA program for organization
    - the impacts and use of EA program and its results
    - the progress and operations of EA program, more specifically the EA team's and architects' accomplishments, the progress of the program towards the pre-established goals and the quality of the results produced by the EA program
    - the architecture structures in organization, i.e. evaluate the architecture alternatives and solutions

    These aspects can be used to support the **identification of the organization's measurement needs and the derivation of appropriate metrics** for each need.

Evaluation Practices

On a more practical level, evaluation practices were studied and especially

- the essential components of evaluation planning were defined to support the evaluation planning activities in organizations. These components include e.g.

    - the objectives set of an EA program,
    - the purpose(s) of evaluation,
    - the targets for evaluation,
    - the users of the evaluation results (audience),
    - quality attributes and metrics,
    - data gathering and analysis techniques, and
    - evaluation process.

    The **maturity** of the organization's EA affects the selection of evaluation targets, as well as the definition of evaluation criteria and metrics. Organizations on lower

levels of EA maturity should start with simple metrics (such as on/off-metrics). While the organization's EA matures, more detailed business impacts can potentially be measured. In addition, the EA evaluation targets and metrics need to be **compatible** with the other evaluation and measurement systems used in the organization (such as Balanced Score Cards).

- a generic evaluation model for EA was constructed to provide a tool to analyze the **status of architecture work** in organizations (which was described earlier in this section). The model consists of three parts:

    - the set of 12 potential CSFs for EA representing the areas to be evaluated,
    - the key questions assigned to each CSF, and
    - the maturity levels to evaluate the status of each CSF.

- the existing architecture evaluation methods were charted, and there seems to be **a lack of methods for evaluating EA**. The most wide-spread approaches at the moment are maturity models and business-IT alignment assessment methods. Since no methods for the evaluation of the entire EA exist, **a combination of methods** is necessary to improve the fulfilment of certain EA evaluation needs. Possible usable methods or techniques can be found e.g. in the areas of business process design, data modelling, software architecture evaluation, benchmark testing, and cost and benefits measurement of ICT investment.

  No matter what methods are used, **EA evaluation depends strongly on conceptual models** as input and the basis for analysis and discussion because the models support sharing and communicating the architectural knowledge among the various stakeholders of architecture work.

Metrics and Criteria for Architecture Evaluation

Because the architecture work is such a large area and many possible evaluation targets exist, the project focused on the following evaluation targets for further scrutiny:

- Architecture documentation: Metrics and criteria were charted for evaluating the quality of architecture documentation. Architecture documentation, descriptions, models etc. of high quality are essential, because they are used as a communication tool and they enable gaining an understanding of the architecture and support effective and succesful implementation of the architecture (plans).

  Architects should ensure the quality of architecture descriptions and models while producing them. Also the quality check of architecture documentation should be included in architecture reviews and quality evaluation checklists should be developed. The metrics and criteria that were charted in our study can be used to support the checklist development in organizations.

- Architecture related communication and commitment to architecture work: A variety of metrics and criteria were charted for evaluating the quality of architecture related communication and commitment. While many organizations are still initializing their architecture programs, the evaluation of communication or commitment, is not considered to have the first priority. However, by identifying the level of satisfaction of a stakeholder e.g. related to the amount and quality of information received through architectural communication, the architecture team can better adjust its communication practices to meet the needs of the stakeholders.

- EA compliance: In general, compliance mainly refers to the conformance with rules (standards, regulations, laws, contracts etc.). In the EA domain, compliance can be divided into internal and external compliance:

    - Internal compliance refers to the compliance between investments (and the projects that implement the investments) and EA with its policies and guidelines.
    - External compliance is about the compliance between EA and business; are the EA guidelines and target state descriptions in line with the business vision, mission, objectives, strategies, and action plans. External compliance may also refer to EA's ability to react to the changing environment of the organization, as well as to the compliance of EA with the laws and regulations the organization needs to obey.

    EA compliance is evaluated in order to

    - direct a project or an investment to comply with EA
    - assure the compliance between the output of a project or an investment and EA
    - assure the compliance between EA and organization's internal or public standards, reference models and principles
    - ensure the usability and appropriateness of EA policies, EA frameworks, EA descriptions, business objectives etc.

    Most typically, the status of EA compliance is illustrated by compliance levels describing whether there is full compliance, partial compliance or non-compliance between e.g. investments and projects, or business and EA.

- Business-IT alignment: Business-IT alignment has been continuously considered as one of the top concerns of company executives. Alignment has also been considered as one of the key benefits or potential objectives of EA.

    Various approaches exist for evaluating business-IT alignment. What seems to differentiate these approaches from each other is that they have a slightly different focus on the issues to be evaluated. A wide selection of evaluation metrics, ranging from soft issues (e.g. communication) to hard issues (e.g. business metrics or skills-related metrics) are provided. It seems that a combination of both qualitative (soft) metrics and quantitative (hard) metrics should be implemented to develop a comprehensive measurement instrument for business-IT alignment. An organization may also need a functional EA governance board which is responsible for evaluating alignment periodically.

- Benefits of architecture work: The need for evaluating the benefits is evident because it provides a rationale for the key stakeholder support and investments in architecture work. In practice, the most typical benefits deal with costs, growth or flexibility. Each of the benefits charted was complemented with up to 60 metrics. As the number of potential metrics is overwhelming, a measurement system for the benefits was proposed.

    The idea is to provide the management of an enterprise with 3-5 metrics which can be used to evaluate architectural work benefits. By using the metrics, the architecture team should rationalize that benefits are received from architectural work in enterprise functions and units. For example, the holistic architectural view of an enterprise, which a high-quality EA can provide, can be used in projects over

and over again, without constructing the architecture separately in the beginning of every project and thus resulting in greater efficiency, speed and accuracy. In addition, management could be also interested in architectural work ROI, because normal investment planning basically applies in architectural work (see e.g. Rosser 2006; Saha 2006; Schmidt 2005).

There are, however, some challenges that relate to architectural work benefit evaluation:

- A baseline or standard for evaluation results does not typically exist in organizations since EA is a new discipline (i.e. no history data yet exists to compare the new evaluation data with).
- It is challenging to find a mutual understanding of the time scale of presenting benefits between management and the architecture team, and a balance between producing short-term and long-term benefits. On one hand, quick wins are essential in gaining management support, but on the other hand, architecture work is long-term in nature.

Even though a vast amount of both qualitative and quantitative metrics for different areas of architecture work were charted, the challenge in the organizations is to **select or derive the most suitable and useful metrics to meet the evaluation needs**, especially in cases where the evaluation needs may not be clear enough. In general, it also seems that metrics selection is dependent on the phase of the architecture development, or more specifically, on the level of architecture maturity: simple metrics (e.g. on-off metrics) may be more usable in the beginning of the EA journey, and more detailed metrics (quantitative and qualitative metrics) may be utilized as the EA work is more established.

Architectural Decision-Making and Risk Management

Finally, the architectural decision-making and risks were briefly addressed to provide a theoretical view to these issues to support the planning of decision-making and risk management practices in organizations.

- Architecture decisions are high level decisions that, in the EA domain, can involve
  - Selection of architecture plans (target, transition, vision)
  - Selection of architecture standards, principles and guidelines
  - Decisions about the objectives of architecture work in the organization.

  It seems that, currently, **EA decisions are not necessarily official or actively made**. The baseline architecture is constantly monitored and improvements planned but the big picture is not necessarily taken into account. The target architecture state is not necessarily officially approved, the architecture transition plans may be merely working papers, and the transitions are not necessarily systematic because of short-term business needs or resource restrictions.

  To be useful, architectural decisions should
  - be made only if absolutely necessary to achieve business strategy and meet architectural objectives
  - be traceable to business objectives
  - not be overly detailed on the enterprise level
  - consider future change needs to the architecture

- be enforceable and enforced
- be communicated with their rationale.

Generally, **architectural decision-making is similar to other kinds of organizational decision-making**. Therefore it should be carefully considered whether or not separate decision-making mechanisms are needed for architecture decisions.

- EA risks can be described as factors that may lead to negative outcomes in the EA program, and as negative outcomes resulting from these factors. Even though **EA-related risks are not currently considered in detail in organizations, there seems to be the need of managing them**. EA risk management is not an independent 'island' in an organization; on the contrary, it should be in a close connection or a part of organizational risk management.

  The overview and categorization of generic risks related to EA can be used to identify typical risks and to assure that risk management practices have been planned for all relevant risks. Additionally, EA risk management may support the attainment of EA objectives, or EA can even be exploited to facilitate organizational risk management.

Summing-up

In the future, architectures and architecture work are quickly becoming an even more important **strategic management tool for organizations** as they seek to rationalize their operations and ICT portfolios as well as alignment between business and ICT. In practice, however, architectural work seems to be very different from theoretical frameworks and process models. Hence, there is **a need for a light and agile EA methodology**, or at least a usable and simple enough EA process, in organizations initiating architectural work. Furthermore, as EA becomes more mature, established practice in organizations, architecture evaluations are required to assess the current status of EA, and to manage and improve it.

Tiivistelmä suomeksi

Tietotekniikan tutkimusinstituutin, TITU:n, toteuttama AISA-projekti on kolmen vuoden ajan tutkinut sekä **kokonais- että ohjelmistoarkkitehtuurien** laadunhallintaa ja arviointia. Näitä kahta keskeistä tutkimusteemaa on tarkasteltu lukuisien tarkempien osateemojen avulla (jotka myös muodostavat tämän julkaisun sisällön).

Ennen projektin tulosten esittelyä on syytä lyhyesti kuvata projektin keskeisimmät käsitteet.

Kokonaisarkkitehtuurin ja ohjelmistoarkkitehtuurin käsitteet

Arkkitehtuureja kehitetään, jotta voidaan kuvata yrityksen tai organisaation rakenne ja osat, joista yritys muodostuu, kuten liiketoimintaprosessit, informaatio, tietojärjestelmät ja tietotekninen infrastruktuuri. Arkkitehtuurien avulla yritysten on mahdollista ymmärtää ja parantaa monimutkaisia rakenteitaan ja sitä kautta myös nopeammin ja tehokkaammin kehittää aidosti liiketoimintaa tukevia tietojärjestelmiä ja reagoida nopeammin liiketoimintaympäristön muutoksiin. Kokonaisarkkitehtuuri (engl. enterprise architecture, EA) ja ohjelmistoarkkitehtuuri (engl. software architecture, SA) sisältävät periaatteessa samoja asioita ja näkökulmia, mutta ohjelmistoarkkitehtuuri

kattaa rajatumman alueen kuin kokonaisarkkitehtuuri.

Lyhyesti sanottuna, **kokonaisarkkitehtuuri, EA, on kokonaisvaltainen lähestymistapa yrityksen tai organisaation hallintaan ja kehittämiseen**. Tyypillisesti EA sisältää seuraavat neljä toisistaan riippuvaa arkkitehtuurista näkökulmaa:

- liiketoiminta-arkkitehtuuri (engl. business architecture),
- tietoarkkitehtuuri (engl. information architecture),
- tietojärjestelmäarkkitehtuuri (engl. information systems architecture), ja
- teknologinen arkkitehtuuri/infrastruktuuri (engl. technology architecture).

Ohjelmistoarkkitehtuuri, SA, taas kuvaa yksityiskohtaisemmin EA:n eri näkökulmiin liittyvää tietoa yhden tarkasteltavana olevan tai suunniteltavan tietojärjestelmän osalta. Niinpä ohjelmistoarkkitehtuuri käsittelee huomattavasti rajatumpaa aluetta kuin EA. Ohjelmistoarkkitehtuuri voidaan nähdä tietojärjestelmien suunnittelun, kehittämisen ja hallinnan työkaluna. Ohjelmistoarkkitehtuuri kuvaa järjestelmän rakenteen osat, osien väliset riippuvuudet, ulkoiset yhteydet sekä kehittäsen olennaisimmat periaatteet.

Seuraavaksi kuvaamme lyhyesti AISA-projektin tulokset tutkimusteemoittain.

Arkkitehtuurityön monitahoisuus

Arkkitehtuurityö on laaja ja monitahoinen alue, joka **kattaa mm. arkkitehtuurien kehittämisen, toteutuksen, hallinnan ja ylläpidon**. Arkkitehtuurityö monitahoisuuden tarkastelun tuloksena AISA-projektissa

- määriteltiin ohjelmistoarkkitehtuurin onnistumisen tai epäonnistumisen tekijöitä sekä kokonaisarkkitehtuurin mahdollisia kriittisiä menestystekijöitä. On selvää, että arkkitehtuurityön onnistumiseen ja laatuun sekä ohjelmisto- että kokonaisarkkitehtuurin tasolla vaikuttavat useat ja toisistaan riippuvat tekijät. Erityisesti

  - viestintä,
  - ylimmän johdon sitoutuminen muiden sidosryhmien (kuten projektipäälliköt tai tietojärjestelmien kehittäjät) sitoutumisen rinnalla sekä
  - arkkitehtuuridokumentaatio (esim. kuvaukset)

  näyttäisivät olevan keskeisimpiä tekijöitä.

  Näiden ja muiden määriteltyjen tekijöiden listaa voidaan yrityksissä käyttää **tarkistuslistana**, jonka avulla voidaan varmistaa, että sekä kokonais- että ohjelmistoarkkitehtuurin kehittäminen ja hallinta on riittävän kattavaa ja suunnitelmallista toimintaa, ja epäonnistumisen mahdollisuus on pieni. Lisäksi kartoitetut tekijät antavat yhden käyttökelpoisen lähtökohdan arkkitehtuurien arviointikriteerien ja mittarien kehittämiselle: ne kuvaavat myös **mahdollisia arviointikohteita**.

- kehitettiin arkkitehtuurityön sidosryhmien viitekehys. Viitekehys sisältää EA:n sidosryhmät ja niiden tyypillisimmät tarpeet. **Arkkitehtuurityöhön eniten vaikuttavien sidosryhmien, kuten ylimmän johdon, tarpeiden huomioiminen ja niihin vastaaminen on ratkaisevan tärkeää, jotta yritys hyötyisi kokonaisarkkitehtuurista**. Rahan ja muiden resurssien lisäksi

sidosryhmät ovat tärkeä tiedonlähde: arkkitehtuurikuvausten tekemiseen tarvittavat tiedot saadaan tyypillisesti sidosryhmiltä.

Koska arkkitehtuurityöllä on paljon mahdollisia sidosryhmiä, luokitellaan nämä viitekehyksessä kolmeen luokkaan, joiden yhteisiä tarpeita myös pohditaan. Lyhyesti sanottuna sidosryhmäroolit voidaan luokitella

- tuottajiin (engl. producers), jotka tekevät arkkitehtuurisuunnittelua ja -kehittämistä
- mahdollistajiin (engl. facilitators), jotka hoitavat arkkitehtuurien hallintaa ja ylläpitoa sekä
- käyttäjiin (engl. users), jotka hyödyntävät arkkitehtuurityön tuloksia/tuotteita päivittäisessä työssään.

Viitekehystä voidaan kättää yrityksessä arkkitehtuurityön **sidosryhmien ja niiden tarpeiden tunnistamiseen**. Sen avulla on myös mahdollista **hallita** paremmin arkkitehtuurityön monien eri sidosryhmien muodostamaa kokonaisuutta.

- kartoitettiin arkkitehtuurityön mahdollisia hyötyjä tukemaan yritysten arkkitehtuurityön ja -ohjelmien **tavoitteiden ja haluttujen hyötyjen määrittelyä**. Yleisimmin oarkkitehtuuriohjelmien avulla halutaan organisaatioissa saavuttaa seuraavia hyötyjä:

  - kustannusten aleneminen,
  - kokonaisvaltainen näkemys yrityksestä ja sen toiminnasta,
  - paremmin liiketoimintaa tukevat tietojärjestelmät ja tekniikka,
  - muutosten parempi hallinta,
  - riskien parempi hallinta,
  - parempi integraatio ja yhteentoimivuus sekä
  - lyhyemmät kiertoajat.

- kartoitettiin arkkitehtuurityön nykytila case-yrityksissä projektissa laaditun alustavan EA:n arviointimallin avulla. Yleisesti ottaen arkkitehtuurityö on yrityksissä vielä melko lailla alkuvaiheissa. Näyttää siltä, että

  - yritykset ovat määritelleet käytettävän arkkitehtuuriviitekehyksen ja arkkkitehtuurisia periaatteita, ja yrityksissä tarjotaan jossakin määrin myös arkkitehtuuriohjausta ICT:tä kehittäville projekteille.
  - arkkitehtuurimallit (kuvaukset) ovat vielä yleensä työn alla, samoin kuin siirtymäsuunnitelma. Siirtymäsuunnitelma kuvaa tarvittavat askeleet, jotta nykytilasta päästään suunniteltuun tavoitetilaan. Nykytilaa ja tavoitetilaa kuvaavia arkkitehtuurimalleja (kuvauksia) on olemassa lähinnä tiettyjen arkkitehtuurinäkökulmien (esim. tietoarkkitehtuurin tai teknisen arkkitehtuurin) osalta, mutta ne eivät välttämättä ole kovin yhtenäisiä eikä kokonaiskuva yrityksen arkkitehtuurista ole yleensä vielä kovin selkeä.
  - arkkitehtuurityön työkalutuki rajoittuu toistaiseksi lähinnä perustoimistotyökaluihin ja yrityksessä jo käytössä oleviin ICT-kehittämistyökaluihin. Näiden työkalujen katsotaan kuitenkin riittävän arkkitehtuuriohjelman käynnistysvaiheessa.
  - arkkitehtuurinen arviointi ja mittaaminen nähdään tärkeänä osa-alueena, mutta arkkitehtuurityö on vielä harvoin yrityksissä sillä tasolla, että arvioinnista koettaisiin saatavan merkittävää hyötyä. Myös käyttökelpoisten arviointikäytänteiden puute aiheuttaa yrityksissä haasteita.

    - arkkitehtuurityö on vielä pikemminkin projekteihin liittyvää toimintaa kuin koko yrityksen tasolla tehtävää systemaattista kehittämistyötä.

- määriteltiin laatuun ja laadunhallintaan liittyviä käsitteitä ja tehtäviä EA- ja SA-alueilla. Projektissa laaditiin **laatumalli sekä ohjelmistoarkkitehtuurien hallinnan prosessille että ohjelmistoarkkitehtuurille** tukemaan laadukkaan ohjelmistoarkkitehtuurin kehittämistä.

  EA-alueella sen sijaan sekä laadukkaan EA:n käsite että EA:n laadunhallinnan tehtävät ovat vielä epäselviä. Lyhyesti sanottuna ehdotamme, että **EA on laadukas mikäli se on ymmärrettävä, hyväksytty ja käytössä yrityksessä, ja EA:ta arvioidaan jotta voidaan varmistua sille asetettujen tavoitteiden ja laatukriteerien täyttymisestä**. EA:n arviointiin on olemassa lukuisia kypsyysmalleja (ks. esim. GAO 2003; IAC 2005; OMB 2005), jotka ovat myös projektissa laaditun EA:n arviointimallin taustalla. Näitä malleja voidaan pitää yhtenä tapana edistää arkkitehtuurityön ja arkkitehtuurien kypsyyttä ja laatua.

Arkkitehtuurityön arviointi

Arkkitehtuurityön arviointiin liittyen AISA-projektissa

- määriteltiin yritysten mahdollisia arviointitarpeita ja arvioinnin taustalla olevia laukaisevia tekijöitä, triggereitä. Arkkitehtuuriarviointeja voidaan käyttää esimerkiksi

    - lisäämään ymmärrystä organisaation liiketoimininta- ja ICT-ympäristöstä sekä taloudellisesta että rakenteellisesta näkökulmasta katsottuna
    - tukemaan muutosten hallintaa, laadunvarmistusta, prosessien suunnittelua ja kehittämistä, IT-kustannusten hallintaa ja arkkitehtuuristen valintojen tekemistä
    - arkkitehtuuridokumentaation laadun, tietorakenteiden, sovellusten ja teknisen infrastruktuurin sekä järjestelmäratkaisujen hallintaan
    - arkkitehtuurityön hallintaan, esimerkiksi varmistamalla että arkkitehtuurityö vastaa odotuksia.

  Yrityksen liiketoimintaan ja ICT:hen liittyvät ongelmat, kysymykset, ajankohtaiset keskustelunaiheet ja tietotarpeet ovat tyypillisimpiä arkkitehtuuriarvioinnin **triggereitä**.

- kartoitettiin lukuisia arkkitehtuurityön arvioinnin näkökulmia. Erityisesti EA-ohjelman arviointi liittyy tyypillisesti johonkin seuraavista näkökulmista:

    - EA-ohjelman hyödyt yritykselle
    - EA-ohjelman vaikutukset, tulokset ja niiden käyttö
    - EA-ohjelman eteneminen ja toiminnot, etenkin EA-tiimin ja arkkitehtien saavutukset ja eteneminen kohti ennaltamääriteltyjä tavoitteita sekä EA-ohjelman tulosten laatu
    - arkkitehtuuriset vaihtoehdot ja ratkaisut

  Kartoitettuja näkökulmia voidaan hyödyntää yrityksissä **organisaatiokohtaisten arviointitarpeiden ja arviointimittareiden määrittelyssä**.

Arvioinnin käytänteet

Käytännönläheisemmällä tasolla AISA-projektissa

- määriteltiin keskeiset arvioinnin suunnittelun komponentit tukemaan arvioinnin suunnittelua yrityksissä. Arvioinnin suunnittelun komponentteihin kuuluvat esimerkiksi

    - EA-ohjelmalle asetetut tavoitteet,
    - arvioinnin tarkoitus (arviointitarpeet ja triggerit),
    - arvioinnin kohteet,
    - arviointitulosten käyttäjät (yleisö),
    - laatuattribuutit/kriteerit ja mittarit,
    - tiedon keräämisen ja analysoinnin tekniikat, sekä
    - arviointiprosessi.

    Yrityksen EA:n **kypsyystaso** vaikuttaa mm. siihen, mitä arvioidaan ja millä kriteereillä ja mittareilla arvioidaan. Yritysten, joissa arkkitehtuurityön kypsyystaso on vielä matala (eli arkkitehtuurityö on vasta alkuvaiheessa), on syytä lähteä liikkeelle yksinkertaisilla mittareilla (kuten ns. kyllä/ei-mittareilla). Kun yrityksen arkkitehtuurityö ja arkkitehtuuri kypsyy, yksityiskohtaisempia mittareita voidaan ottaa käyttöön. Lisäksi on syytä huomata, että EA:n arvioinnin kohteiden ja mittareiden tulisi olla **yhteensopivia** yrityksessä käytössä olevien muiden mittareiden tai mittausohjelmien, esimerkiksi tuloskorttien (engl. Balanced Score Card), kanssa.

- laadittiin yleinen EA:n arviointimalli, jonka avulla voidaan analysoida **yrityksen arkkitehtuurityön nykytila**. Malli koostuu kolmesta osasta:

    - kahdentoista mahdollisen EA:n kriittisen menestystekijän joukko edustaa arkkitehtuurityön arvioitavia osa-alueita,
    - jokaiseen menestystekijään liittyvät keskeiset kysymykset auttavat kartoittamaan kyseisen osa-alueen tilannetta ja
    - kuusiportaisen kypsyysasteikon avulla kullekin osa-alueelle annetaan numeerinen kypsyysarvio.

- kartoitettiin olemassa olevia arkkitehtuurien arviointimenetelmiä. Näyttää siltä, että erityisesti **EA-alueella menetelmistä on pulaa**. EA-arvioinnissa yleisimmin käytettyjä lähestymistapoja tällä hetkellä ovat kypsyysmallit ja liiketoiminnan ja IT:n yhteensovittamista (engl. business-IT alignment) arvioivat mallit. Koska toistaiseksi ei näytä olevan tarjolla yhtä yksittäistä menetelmää kokonaisarkkitehtuurin kattavaan arviointiin, joudutaan käyttämään useamman menetelmän **yhdistelmää**, jotta voidaan täyttää monet erilaiset EA:n arvioinnin tarpeet. Käyttökelpoisia arvioinnissa hyödynnettäviä menetelmiä tai tekniikoita voi lötyä esimerkiksi prosessien kehittämisen, tietojen mallinnuksen, ohjelmistoarkkitehtuurien arvioinnin, benchmark-testauksen ja ICT-investointien kustannusten ja hyötyjen arvioinnin alueilta.

    Huolimatta siitä, mitä arviointimenetelmiä tai -tekniikoita yrityksissä käytetään, **EA:n arviointi nojautuu suurelta osin käsitteellisiin malleihin (kuvauksiin)**; mallit toimivat sekä arvioinnin syötteenä että analysoinnin ja keskustelun pohjana. Arkkitehtuurimallit tukevat arkkitehtuurisen tiedon jakamista ja viestintää eri sidosryhmien välillä.

Arkkitehtuurityön arvioinnin kriteerit ja mittarit

Koska arkkitehtuurityö on laaja alue, on mahdollisia arviointikohteitakin useita. AISA-projektissa rajauduttiin tarkastelemaan seuraavia arviointikohteita:

- Arkkitehtuuridokumentaatio: Mittareita ja kriteereitä arkkitehtuuridokumentaation - kuvausten, mallien jne. - laadun arviointiin kartoitettiin laajalti. Laadukas arkkitehtuuridokumentaatio on olennaista, koska kuten edellä jo mainittiin, niitä käytetään kommunikointivälineenä, ne edesauttavat arkkitehtuuriymmärryksen saavuttamista ja tukevat myös arkkitehtuurien menestyksekästä toteuttamista suunnitelmien mukaan.

  Arkkitehtien tulisi varmistaa arkkitehtuuristen kuvausten ja mallien laatu jo näitä dokumentteja tuotettaessa. Arkkitehtuuristen dokumenttien laaduntarkistus tulisi myös sisällyttää arkkitehtuurikatselmointeihin (engl. architecture review) ja laadun arvioimiseksi tulisi kehittää tarkistuslistoja. Mittareita ja kriteereitä, joita projektissa kartoitettiin, voidaan hyödyntää yrityksissä näiden tarkistuslistojen laatimisessa.

- Arkkitehtuurinen viestintä ja sitoutuminen arkkitehtuurityöhön: Joukko mittareita ja kriteereitä kartoitettiin arvioimaan arkkitehtuurisen viestinnän ja sitoutumisen laatua tai tasoa. Koska monet yritykset ovat vasta käynnistämässä arkkitehtuuriohjelmiaan, ei viestinnän tai sitoutumisen mittaaminen ole tärkeysjärjestyksessä ensimmäisenä. Eri sidosryhmien mielipiteiden ja tyytyväisyyden arviointi (esim. viestinnän keinoin saatuun arkkitehtuureihin liittyvän tiedon määrään tai laatuun liittyen) voi kuitenkin auttaa arkkitehtitiimiä tai -yksikköä sovittamaan viestintäänsä vastaamaan paremmin sidosryhmien tarpeita.

- EA:n mukaisuus, EA:n noudattaminen (engl. EA compliance): Yleisesti ottaen jonkin mukaisuus viittaa pääasiassa tiettyjen sääntöjen (kuten standardien, lakien tai sopimusten) noudattamiseen. AISA-projektissa havaittiin, että EA:n mukaisuus, EA:n noudattaminen, voidaan jakaa sisäiseen ja ulkoiseen noudattamiseen:

  - Sisäinen EA:n noudattamisen aspekti (engl. internal compliance) liittyy siihen, että yrityksen investoinnit ja projektit noudattavat EA:ta ja sen periaatteita ja ohjeita.
  - Ulkoinen EA:n noudattamisen aspekti (engl. external compliance) viittaa puolestaan EA:n ja liiketoiminnan väliseen yhteyteen ja yhdenmukaisuuteen; ovatko EA:n ohjeet ja tavoitetilakuvaukset yhteneviä liiketoiminnan vision, mission, tavoitteiden, strategioiden ja toimintasuunnitelmien kanssa. Ulkoinen aspekti voi viitata myös EA:n kykyyn reagoida yrityksen toimintaympäristön muutoksiin sekä siihen, noudattaako EA niitä lakeja ja säännöksiä, joita yrityksen tulee noudattaa.

  EA:n mukaisuutta on tarpeen arvioida, jotta voidaan

  - ohjata projektia tai investointia noudattamaan EA:ta ja sovittuja pelisääntöjä
  - varmistaa, että projektin tai investoinnin lopputulos vastaa suunniteltua EA:ta
  - varmistaa, että EA noudattaa yrityksen käyttämiä sisäisiä tai julkisia standardeja, referenssimalleja tai periaatteita
  - varmistaa, että EA:n periaatteet, viitekehys, kuvaukset, liiketoiminnan

tavoitteet jne. ovat käyttökelpoisia ja tarkoituksenmukaisia.

Tyypillisesti EA:n noudattamisen taso arvioidaan kolme- tai useampiportaisella asteikolla; investointi tai projekti voi olla esimerkiksi kokonaan (engl. full compliance), osittain (engl. partial compliance) tai ei ollenkaan (engl. non-compliance) EA:n ja sen periaatteiden mukainen.

- Business-IT alignment (BIA), liiketoiminnan ja IT:n yhteensovittaminen: Liiketoiminnan ja IT:n yhteensovittaminen on jo pitkään ollut keskeinen kehittämisen tavoite niin IT-johdon kuin yritysjohdonkin mielestä. Se on myös yksi EA:n tavoite ja onnistuneen arkkitehtuurityön aikaan saama hyöty.

  BIA:n arviointiin on olemassa useita erilaisia lähestymistapoja, joita erottaa lähinnä erilaiset arvioinnin painopistealueet tai kohteet. Tarjolla on myös suuri joukko sekä ns. pehmeitä mittareita (esim. viestintään liittyen) että kovia mittareita (esim. liiketoimintaan ja osaamiseen liittyen). Näyttäisi siltä, että BIA:n arvioimiseksi tarvitaan yhdistelmä sekä laadullisia (pehmeitä) mittareita että määrällisiä (kovia) mittareita. Lisäksi voi olla tarpeen hyödyntää esimerkiksi EA:n johtoryhmää (engl. EA governance board) tai vastaavaa toimielintä, joka vastaisi BIA:n arvioinnista säännöllisesti.

- Arkkitehtuurityön hyödyt: Hyötyjen arvioimiselle on yrityksissä selkeä tarve. Arvioinnin avulla voidaan osoittaa eri sidosryhmille arkkitehtuurityön hyödyt ja perustella siten siihen käytettävä panostus. Tyypillisimmin hyödyt liittyvät joko kustannuksiin, kasvuun tai joustavuuteen (ketteryyteen). Projektissa kartoitettuihin hyötyihin liittyi myös suuri joukko mittareita, lähes 60. Koska mahdollisia mittareita on niin paljon, laadittiin ehdotus hyötyjen mittausjärjestelmäksi.

  Perusidea on, että yrityksen johdolle tarjotaan 3-5 mittaria, joiden avulla arkkitehtuurityön hyötyjä voidaan arvioida. Näiden mittareiden avulla arkkitehtitiimin tulisi kyetä perustelemaan, että hyödyt on saatu aikaan yrityksen toimintoihin ja yksiköhin vaikuttaneen arkkitehtuuritön tuloksena. Esimerkiksi kokonaisvaltainen näkemys yritykseen ja sen toimintaan, jonka laadukas EA siis mahdollistaa, on käyttökelpoinen apuväline kaikissa yrityksen projekteissa eikä kokonaisarkkitehtuurin tai sen osa-alueen mallia tai kuvausta tarvitse jokaisen projektin alussa erikseen laatia. Tämän pitäisi näkyä sekä tehokkuuden, nopeuden että tarkkuuden kasvuna projekteissa. Lisäksi yrityksen johto voi olla kiinnostunut arkkitehtuurin tuottoasteesta (engl. architectural ROI), koska investointisuunnittelun käytänteitä voidaan soveltaa myö arkkitehtuurityöhön (ks. esim. Rosser 2006; Saha 2006; Schmidt 2005).

  Arkkitehtuuritön hyötyjen arvioinnissa on kuitenkin myös haasteita:

  - Vertailukohtaa, historiatietoja tai raja-arvoja arviointitulosten analysointiin ei yleensä yrityksissä ole olemassa, koska arkkitehtuurityö on vielä varsin uutta toimintaa eikä ensimmäistä kertaa tehtävän arvinoinnin tuloksia voida siten verrata aikaisempaan tilanteeseen.
  - Yhteisen ymmärryksen löytäminen siitä, millä aikajänteellä hyötyjä tulisi arvioida on haastavaa; yrityksen johto yleisesti ottaen haluaa nähdä nopeasti hyötyjä, arkkitehtitiimi katsoo asioita pitemmällä tähtäimellä. Yrityksissä joudutaankin tasapainoilemaan nopeiden voittojen ja pitkän aikavälin hyötyjen osoittamisen välillä. On syytä huomata, että nopeiden voittojen osoittaminen on tärkeää yritysjohdon tuen saamiseksi ja vahvistamiseksi,

mutta arkkitehtuurityö sinänsä on luonteeltaan pitkän tähtäimen toimintaa.

Vaikka projektissa kartoitettiin suuri joukko laadullisia ja määrällisiä mittareita arkkitehtuurityön eri osa-alueiden arviointiin, haasteena yrityksissä on **valita tai johtaa niistä omiin tarpeisiinsa soveltuvimmat mittarit**, erityisesti sellaisissa tapauksissa, joissa mittaamisen tarpeet eivät ole vielä riittävän selkeitä. Yleisesti ottaen näyttää myös siltä, että arviointimittareiden valintaan vaikuttaa arkkitehtuurityön vaihe, tarkemmin sanoen arkkitehtuurityön kypsyys: yksinkertaisemmat mittarit (esim. ns. kyllä/ei-mittarit) lienevät käyttökelpoisempia arkkitehtuurityön alkuvaiheissa ja yksityiskohtaisempia mittareita (sekä laadullisia että määrällisiä) kannattanee käyttää kun arkkitehtuurityöstä on tullut jo vakiintuneempaa toimintaa yrityksessä.

Arkkitehtuurinen päätöksenteko ja riskien hallinta

Arkkitehtuuriseen päätöksentekoon ja arkkitehtuurisiin riskeihin projektissa luotiin lähinnä teoreettiset katsaukset, joiden tarkoituksena on osaltaan tukea yrityksiä päätöstentekoon ja riskien hallintaan liittyvien käytänteiden suunnittelussa.

- Arkkitehtuuriset päätökset ovat korkean tason päätöksiä, jotka esimerkiksi EA-alueella voivat liittyä

  - arkkitehtuurisuunnitelmien (tavoitearkkitehtuuri, siirtymäsuunnitelma, arkkitehtuurivisio jne.) valintatilanteisiin
  - arkkitehtuuristandardien, periaatteiden, suuntaviivojen tai ohjeiden valintatilanteisiin
  - yrityksen arkkitehtuurityön tavoitteisiin.

  Näyttää siltä, että **EA-päätöksiä ei tällä hetkellä tehdä yrityksissä välttämättä virallisesti tai aktiivisesti**. Olemassa olevan arkkitehtuurin osa-alueita voidaan kyllä ottaa huomioon ja kehityssuunnitelmiakin niille laaditaan, mutta arkkitehtuurin kokonaiskuvaa ei välttämättä hyödynnetä. Tavoitearkkitehtuuria ei ehkä ole virallisesti hyväksytty yrityksessä eivätkä siirtymätkään kohti tavoitetilaa ole kovin systemaattisia johtuen siirtymän laukaisevista lyhyen tähtäimen liiketoimintatarpeista tai siitä, että resursseja ei sillä hetkellä ole saatavilla riittävästi pitkällä tähtäimellä edullisemman ratkaisun toteuttamiseen.

  Jotta arkkitehtuurisista päätöksistä olisi jotain hyötyä,
  - niitä pitäisi tehdä vain jos ne ovat välttämättömiä arkkitehtuuristen tavoitteiden saavuttamiseksi
  - niiden tulisi olla jäljitettävissä liiketoiminnan tavoitteisiin
  - niiden ei tulisi olla liian yksityiskohtaisia
  - niiden tulisi ottaa huomioon mahdolliset arkkitehtuurin muutostarpeet tulevaisuudessa
  - niiden tulisi olla toteuttamiskelpoisia ja ne tulisi myös toteuttaa
  - ne tulisi kommunikoida ja perustella asiaankuuluville sidosryhmille.

  Yleisesti ottaen **arkkitehtuuriset päätökset eivät juuri eroa muusta päätöksenteosta**, joten yrityksissä on tarkkaan harkittava onko tarvetta ottaa niille käyttöön erillisiä päätöksentekomekanismeja.

- EA-riskit voidaan määritellä tekijöiksi, joilla on negatiivinen vaikutus EA-ohjelman tuloksiin ja toisaalta negatiivisiksi tuloksiksi, joita näistä tekijöistä aiheutuu. Vaikkakaan **EA:han liittyviä riskejä ei yrityksissä tällä hetkellä**

syvällisesti pohdita, näyttää siltä, että niiden hallinnalle löytyisi tarvetta. EA-riskien hallinta ei ole kuitenkaan mikään erillinen, itsenäinen saareke yrityksissä, vaan sen tulee olla läheisesti kytköksissä organisaation muuhun riskien hallintaan tai jopa osa sitä.

Projektissa tehtyä yleiskatsausta EA:n riskeistä ja niiden luokittelusta voidaan käyttää yrityksissä tukemaan tyypillisimpien riskien tunnistamista ja varmistamaan, että riskien hallinnan käytänteet kattavat kaikki relevantit riskit. Lisäksi EA:n riskien hallinnan avulla voidaan tukea EA:n tavoitteiden saavuttamista tai EA voidaan nähdä myös yhtenä keinona edistää ja tukea yrityksen riskien hallintaa.

Yhteenveto

Arkkitehtuurit ja arkkitehtuurityö näyttävät tulevan yhä tärkeämmiksi yritysten **strategisiksi työkaluiksi**, koska niiden avulla yritykset pyrkivät järkeistämään toimintojaan ja ICT-porftolioitaan sekä kehittämämään IT:tään liiketoimintalähtöisesti, tavoitellen siis paremmin liiketoimintaa tukevia tietojärjestelmiä. Käytännön arkkitetehtuurityö näyttää kuitenkin olevan melko kaukana siitä kuvasta, jonka teoreettiset viitekehykset ja prosessimallit arkkitehtuurityöstä antavat. Niinpä tarvetta on edelleen **kevyelle ja ketterälle EA-menetelmälle** tai ainakin helppokäyttöiselle ja riittävän yksinkertaiselle EA-prosessille tukemaan yrityksissä käynnistettävää arkkitehtuurityötä. Kun EA ja arkkitehtuurityö yrityksissä on kypsempää ja vakiintuneempaa toimintaa, tarvitaan myös systemaattista arkkitehtuurien ja arkkitehtuurityön arviointia EA:n hallinnan ja kehittämisen tueksi.

Top of page Page Index

Authors: Eetu Niemi & Tanja Ylimäki

AISA Project

In this section, the following topics are covered:

Project Themes and Objectives

In this project conducted by the Information Technology Research Institute (ITRI), we focused on studying the quality management and evaluation aspects of both **enterprise architecture (EA)** and **software architecture (SA)**. Architectures are built to depict the structure of an enterprise and its components, such as processes, information systems (IS), information and IT platforms. They enable enterprises to understand and improve their complex structures, and thus more quickly and efficiently develop information systems that truly support the business, and react to changes in the business environment. EA and SA share the same fundamental idea but differ in scope.

To put it briefly, EA is a holistic approach for managing and developing whole enterprises and typically describes the following four interdependent architectural dimensions:

- business architecture,
- information architecture,
- application architecture, and
- technology architecture.

SA, on the other hand, depicts information from all of the dimensions of EA in more details from the viewpoint of information systems. Thus, it has significantly narrower scope than EA. SA is a tool for planning, developing and managing software-intensive systems. It describes the components of a system, their interrelationships, external connections, and essential principles of development. Typically, SA includes the following viewpoints:

- functional
- information
- development / external
- deployment
- operational.

More precise definitions of EA and SA can be found in the section of Enterprise and Software Architecture Work.

The AISA project focused on **two main research questions**:

- What are the characteristics of architecture planning and development **processes** of high quality and maturity?
- What are the characteristics of enterprise and software **architectures** of high

quality and maturity?

To tackle these questions, research and development in the AISA project was divided into several themes. Research was conducted during three years (2005-2008). In the following, the areas studied in each year are presented.

During the first year of the project the focus was on

- architecture success from the viewpoint of architecture maturity and quality,
- architecture quality management processes, and
- architecture work status and development needs in ICT-provider and user organizations.

During the second year of the project we studied

- architecture quality evaluation criteria and metrics, and
- architecture quality management/evaluation methods and practices.

The last project year dealt with

- architectural decision-making and
- architectural risks.

The **primary outcomes** of the project consist of (see Results section for more details)

- success factors for EA and SA
- quality management activities for EA and SA
- current status of architecture work in companies
- stakeholders and benefits of EA
- role of architecture evaluations in organizations
- EA evaluation planning components
- current state of EA evaluation methods and practices
- metrics for evaluating architectures and architecture processes, especially
    - architecture documentation,
    - communication and commitment, and
    - architecture benefits
- aspects of compliance and business-IT alignment in the EA context
- architecture related decision-making and risk management.

Top of page

Research Approaches and Methods

The project utilized the **design science paradigm** (see e.g. Hevner et al. 2004). According to this paradigm, information from both literature and the real world is combined by researchers to build theories and artifacts which are then evaluated. Design science research contributes to both research and practice via additions to the scientific knowledge base and practical applications. For such a novel topics as EA and SA, this paradigm was considered feasible since only few established theories and models exist, and a great amount of the knowledge is situated in the practical domain.

Following the design science paradigm, both empirical and theoretical data was utilized in the project. The most used means of data collection were literature review and focus group interview. Literature reviews were carried out systematically. In a typical literature review in the project, a keyword search was first carried out in four high-quality academic databases (Academic Search Elite, Electronic Journals Service, Science Direct and Web of Science) and Google Scholar using typical keywords related to the research topic in question. A preliminary set of potentially relevant literature was identified by this search. Subsequently, the found literature was charted for references, and forward and backward search (see e.g. Levy and Ellis 2006) utilized to obtain deeper and wider literature background.

Focus group interviews (see e.g. Krueger and Casey 2000) in the project were typically carried out after literature reviews to

- to validate the literature review results, and
- to collect additional, experience-based information.

Generally two persons from each of the participating companies were invited to the interview. The company representatives were allowed to invite the most suitable interviewees for each of the interview topics as they had the best knowledge about the fields of know-how of their personnel. Group interview was considered a feasible method of data collection, because group influence was thought to stimulate the discussion and thus bring out as much information as possible related to the novel topics covered in the project. However, confidential information may have remained undisclosed for the same reason.

The focus group interviews were moderated by one researcher, while the other one or two took notes. In addition to the notes taken, the interviews were audio-recorded. The duration of a focus group interview on one research topic in the project was from two to three hours. Each interview was succeeded by an analysis and consolidation phase, were the results from the literature review and the focus group interview were combined and examined against each other.

In the company-specific research on the status of architecture work in companies, semi-structured interviews were carried out to collect company-specific data. In these interviews, a guiding interview framework constructed according to literature was utilized to structure and analyze the results. From one to three architecture experts were interviewed in each company.

Top of page

Organization

Project conducted by:	Information Technology Research Institute (ITRI), University of Jyväskylä
Duration:	1.2.2005 - 31.3.2008
Funding:	• The Finnish Funding Agency for Technology and Innovation (Tekes) • The participating companies
Project Board:	• Markku Sakkinen (University of Jyväskylä), Scientific Director (1.2.2005 - 31.8.2006) • Kari Kärkkäinen (ITRI, Scientific Director (1.9.2006 -

	1.4.2007) • Hannakaisa Isomäki (ITRI), Scientific Director (1.4.2007 - 31.3.2008) • Niina Hämäläinen (ITRI), Project Manager (1.2.2005 - 31.12.2007) • Eetu Niemi (ITRI), Project Manager (1.1. - 31.3.2008) • Jaakko Viitanen (A-Ware) • Ari Andersin (Elisa) • Petri Ahveninen (IBM) • Stina Carlsson (IBM) • Jouko Poutanen (IBM) • Markku Korhonen (OPK) • Tapani Vertanen (SOK) • Sakari Olli (Tieturi) • Jari Kovanen (Tekes) • Timo Taskinen (Tekes)
Project team:	The following researchers have participated in the project: • Martin Hoffman (1.1. - 31.6.2007) • Niina Hämäläinen (1.2.2005 - 10.2.2008, Project Manager 1.2.2005 - 31.12.2007) • Denis Kozlov (1.5. - 30.6.2006) • Eetu Niemi (1.2.2006 - 31.3.2008, Project Manager 1.1. - 31.3.2008) • Tanja Ylimäki (1.2.2005 - 11.3.2008)
Participating companies:	• A-Ware Oy - www.aware.fi • Elisa Oyj - www.elisa.fi • IBM Finland - www.ibm.com/fi • Osuuspankkikeskus, OPK (OP Bank Group Central Cooperative) - www.op.fi • SOK (S Group) - www.sok.fi • Tieturi - www.tieturi.fi

Results - papers, reports, and presentations - can be found in the Results section, and the Lessons Learned section covers the discussion of the project outcome.

Top of page Page Index

Author: Niina Hämäläinen, Eetu Niemi and Tanja Ylimäki

Enterprise and Software Architecture Work

In this section, we deal with the following topics:

- Architecture Approaches: Brief description of the various architecture apporaches used in organizations
- Definitions for Enterprise and Software Architecture: Overview and definition of the concepts of EA and SA
- Potential Critical Success Factors for Enterprise Architecture: Overview and description of the factors enabling the success of EA
- Success and Failure Factors for Software Architecture: Overview and description of the factors that enable or prevent the success of SA
- Stakeholders of Architecture Work: Overview and description of the stakeholders of EA
- Benefits of Architecture Work: Overview and description of the benefits of EA
- Status of Architecture Work: Overview and description of the status of arhitecture work in organizations

Architecture Approaches in Organizations

Organizations develop their architecture capabilities based on different architecture management approaches. Currently, a variety of approaches for architecture management are studied, developed and applied both by industry and academia. Examples of these are:

- Enterprise Architecture, EA
- Information System Architecture
- Software Architecture, SA
- Integration Architecture
- Technology Specific Architectures (such as J2EE, .Net)
- Service Oriented Architecture, SOA
- Information / Data Architecture
- Process Architecture
- Computer Architecture

Each of these architecture management approaches has its particular scope, emphasized aspects, and architecting activities. Some of the approaches cover a wide part of organization's structures and describe those in a high level (specifically EA). Others focus on particular area and aspects and describe these in more detailed levels (e.g. SA). Furthermore, the utilization of the results (such as enterprise models) produced by these different approaches varies. Some results can be used in the company management and some results are used to support the design and development of information systems.

While there is no single agreement in the industry or in academia on the meaning of these architecture management approaches or on their relationship to one another, an organization may handle the confusing situation, for example, by being

> "aware that these different terms exist, but that there is no consistent definition of these terms in the industry and how they relate. The recommendation, therefore, is for you to select the terms relevant to your organization and define them appropriately. You will then achieve some consistency at least and reduce the potential for miscommunication". (Eeles,

2006b)

More information on this subject can be found in the dissertation Evaluation and Measurement in Enterprise and Software Architecture Management.

Top of page

Definitions for Enterprise and Software Architecture

In AISA project, the focus was on enterprise and software architecture approaches. Next, we will define these concepts.

Enterprise Architecture: In the modern turbulent business environment, companies are constantly encountering challenges in coping with the changes and complexity in the market. Moreover, the companies have to manage the complexity of their information and communication technology (ICT) environment brought on by the many decades long legacy of ICT, and to assure that ICT supports the business as well as possible. To facilitate companies in responding to these challenges, a recent approach called Enterprise Architecture (EA) has emerged in the last decade (Goethals et al. 2006; Hjort-Madsen 2006; Kluge et al. 2006; Morganwalp & Sage 2004; Veasey 2001). Consequently, the approach has become one of the major concerns of practitioners and academics, and it is being implemented in a multitude of companies and government organizations worldwide.

Basically, EA is a holistic approach for managing and developing an organization, adopting an overall view of its business processes, information systems (IS), information and technological infrastructure (de Boer et al. 2005; Jonkers et al. 2006; Kaisler et al. 2005). EA includes a set of principles, methods and models used to describe the current and future state of an organization, as well as a transition plan to describe the steps needed to transform from the current to the target state (Armour et al. 1999; Lankhorst 2005). The transformation is usually conceptualized as a continuous, iterative process (Armour et al. 1999; Kaisler et al. 2005; Pulkkinen & Hirvonen 2005).

One of the most promising definitions of EA is the following:

> "Enterprise architecture (EA) identifies the main components of the organization, its information systems, the ways in which these components work together in order to achieve defined business objectives, and the way in which the information systems support the business processes of the organization. The components include staff, business processes, technology, information, financial and other resources, etc. Enterprise architecting is the set of processes, tools, and structures necessary to implement an enterprise-wide coherent and consistent IT architecture for supporting the enterprise's business operations. It takes a holistic view of the enterprise's IT resources rather than an application-by-application view." (Kaisler et al., 2005)

Software Architecture: SA, on the other hand, depicts information from all of the dimensions of EA in more detailed form from the viewpoint of information systems. Thus, it has significantly narrower scope than EA. SA a tool for planning, developing and managing software-intensive systems.

SA has emerged as the principled understanding of the large-scale structures of software systems (Shaw & Clements, 2006). It offers guidance for complex software design and

development (Shaw & Clements, 2006). Software architecture management is utilized in the information system development and it can be defined as follows:

> "The software architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them." (Bass et al., 2003)

Typically, SA includes the following viewpoints:

- Functional
- Information
- Behavioral / Concurrency
- Development / External
- Deployment
- Operational

Architecting of a system contributes to the development, operation, and maintenance of a system from its initial concept until its retirement from use (ISO, 2007). Rozanski and Woods (Rozanski & Woods, 2005) describe architecture development as a process by which stakeholder needs and concerns are captured, an architecture to meet these needs is designed, and the architecture is clearly and unambiguously described via an architectural description. Architecting of software architecture consists of activities of

- capturing architectural requirements of system/software and understanding them,
- designing and analyzing/evaluating,
- realizing, maintaining, improving, and
- certifying the architecture as well as documenting it (IEEE, 2000, Bass et al., 1998).

In addition, the co-operation with EA planning relate to software architecting, especially communicating and giving feedback about enterprise-wide principles.

More information on this subject can be found in the dissertation Evaluation and Measurement in Enterprise and Software Architecture Management.

Top of page

Potential Critical Success Factors for Enterprise Architecture

The concept of critical success factor (CSF) has been utilized in Total Quality Management (TQM) to indicate those issues that must be done exceedingly well in order to succeed (Badri et al., 1995, Claver et al., 2003; Lecklin, 2002; Tarí, 2005). While the idea of CSF has later on found its way to many other areas as well (such as project management), it awakened our interest for studying the CSFs in the context of EA: what are the factors that have to be carried out exceedingly well in order to attain a successful EA – a high-quality EA – which in turn enables the business to reach its objectives and gain more value.

In our study, we determined the potential CSFs for EA (see figure and table below), more

specifically a set of potential key areas from which the organization should choose the most critical factors of its own based on its business objectives, the role of EA in the organization, and so forth. These factors, when carefully addressed, should enable the achievement of a high-quality EA.

Figure: Potential critical success factors for EA.

Table: Brief descriptions of the potential critical success factors for EA.

CSF for EA	**Description**
Assessment and Evaluation	The extent to which the architecture and architecture processes are evaluated and improved, and how established the evaluation processes are. Deals with issues such as definition of EA evaluation
Business Driven Approach	The extent to which the business strategies, business objectives and requirements are taken into account in the architecture development.
Commitment	The extent to which both the top-management and the employees of the organization are committed to and involved in the EA effort.
Communication and Common Language	The extent to which the organization has established architecture related terminology (the common vocabulary) and effective means to conduct architecture related communication.
Development Methodology and Tool Support	The extent to which the organization has an established architecture framework and development process, and the extent to which different tools are exploited in architecture development and management.
EA Models and Artifacts	Deals with issues such as developing a documentation plan, collecting and analyzing the requirements, ensuring that all necessary views are modeled in order to provide a coherent and concise picture of the enterprise (current and future models), and developing a transition plan.
	Relates to issues such as governance (architecture guidance) structures, roles, responsibilities, processes

Governance	and activities, change management processes (both organizational and architectural changes) and risk management processes.
IT Investment and Acquisition Strategies	Deals with the relationship (and dependency) between architecture development and governance processes and the IT investment and acquisition processes and decisions.
Organizational Culture	Deals with issues such as the organization's readiness to develop and utilize EA, attitudes towards the architecture approach, attitudes towards changes in general, and the organizational changes the architecture development may lead to.
Project and Program Management	Deals with issues such as the coordination between various (architecture) projects, utilization of project milestones and checkpoints for architectural evaluation or guidance, taking advantage of lessons learned and best practices, as well as being on budget and schedule.
Scoping and Purpose	Deals with issues such as the definition of EA in the organization, the key stakeholder groups, the mission, goals and direction of EA, the purpose of EA, and how wide organizationally, how deep and detailed and how fast the EA should be developed in the organization.
Skilled Team, Training and Education	The extent to which the architecture team is organized and established as well as the extent to which required skills are available or acquired.

For more information on CSFs for EA, see either the related paper or report.

Top of page

Success and Failure Factors for Software Architecture

Software architecture is a critical factor in the design and construction of any complex software-intensive system. SA impacts on the quality of the system. A good architecture can help ensure that a system will satisfy key requirements in such areas as performance, reliability, portability, scalability, and interoperability (Garlan, 2000).

Architecture evaluation is a way to increase the understanding of the quality of architecture. A variety of methods is being developed for the evaluation of software architectures, for example

- SAAM (Babar, Zhu et al., 2004),
- ATAM (Bass, Clements et al., 1998),
- ARID (Bengtsson, Lassing et al., 2004), and
- ALMA (Boehm, 1994).

There seem to be no consensus on the evaluation objectives, criteria, or evaluation targets in the SA literature. In some method comparisons evaluation objectives and use cases are discussed (e.g. Clements, Kazman et al., 2002). Also the evaluation criteria and metrics are still unestablished; several evaluation criteria and metrics descriptions exist. SA

evaluation criteria are discussed for example byHilliard et al. (1996; 1997) and Losavio et al. (2003; 2004). One reason for the non-establishment of architecture evaluation criteria and metrics may be that no common views on what is successful SA and what factors have an effect on achieving successful SA exist. It is not clear what targets and factors should be evaluated and measured. Nevertheless, successful architecture is a widely used concept.

Academia and practitioners have come to realize that a critical success factor for system design and development is to find a successful architecture. They have also become increasingly interested in what makes SA succeed or fail. The identified success and failure factors help system development managers and architects make a number of critical decisions. These decisions relate, for example, to the selection of evaluation criteria and metrics for the quality assessment of architectures and architecture management processes.

It is generally known that the success of SA is typically influenced by factors at various levels. However, these factors are mainly discussed only in a few studies and reports organised and produced by some research institutes and the ICT industry (e.g. Avritzer and Weyuker, 1999; van der Raadt, Soetendal et al., 2004). These factors are, as yet, far from having been fully investigated in detail. Our study contributed to this field with an identification and analysis of success and failure factors of SA.

We identified six system development areas that seem to affect the success and failure of SA (see the figure below).

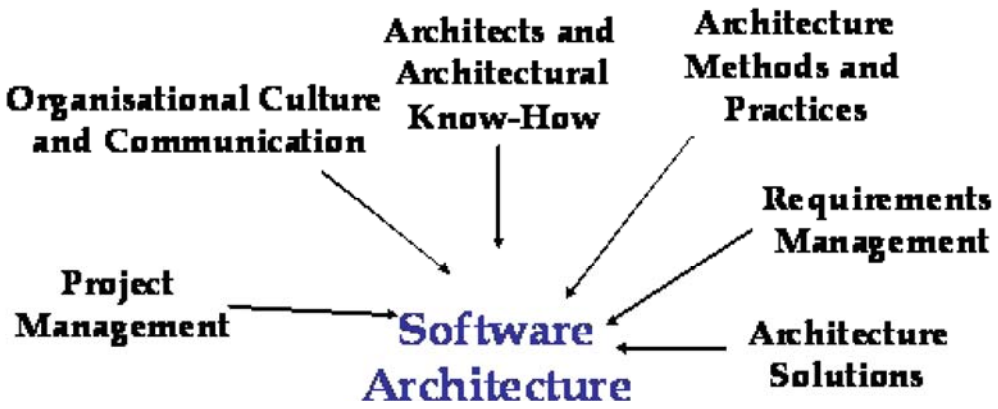


Figure: System development areas affecting the success and failure of software architecture.

Project management offers time, staff and resources for architectural work. Software architecture success factors relating to the project management can be divided into factors relating to staffing, scheduling, planning and funding.

Organizational culture refers to the values, beliefs and customs of an organization. Whereas organizational structure is relatively easy to draw and describe, organizational culture is less tangible. Organizational culture has an impact, for example, on how well the architecture will be adopted and followed.

Architects and Architectural Know-How: The personal skills of architects have an effect on the fluency of the architectural design process in collaboration with the stakeholders. Personal skills may also have an impact on architectural decision making.

Architecture Methods and Practices: The software architecture management process contributes to the activities of capturing architectural requirements and understanding them, designing, analyzing/evaluating, realizing, maintaining, improving, and certifying the architecture as well as documenting it [3, 14]. The process model together with the methods and tools chosen to carry out architectural work, in turn have influence on this work. In addition, the standardization of the architectural concepts and of the descriptions in an organization has an effect on the architectural practices.

Requirements management: Architectural design and decision making is founded on identified requirements. Previous studies do not clearly highlight which factors in the requirements management advance the success of software architecture. However, the problems in requirements quality cause failure for software architecture like as described in the next chapter. Therefore, it is evident that the quality of the requirements and of the requirements management process advances the success of software architecture.

Architecture solutions: Architectural choices and decisions are made in architectural design. Based on these decisions, the architectural specifications are produced.

The main success factors and their relationship are presented in the figure below.

Similarly, failure factors for SA were defined for the different areas. Examples of both success and failure factors for SA are presented in the table in the end of this topic.

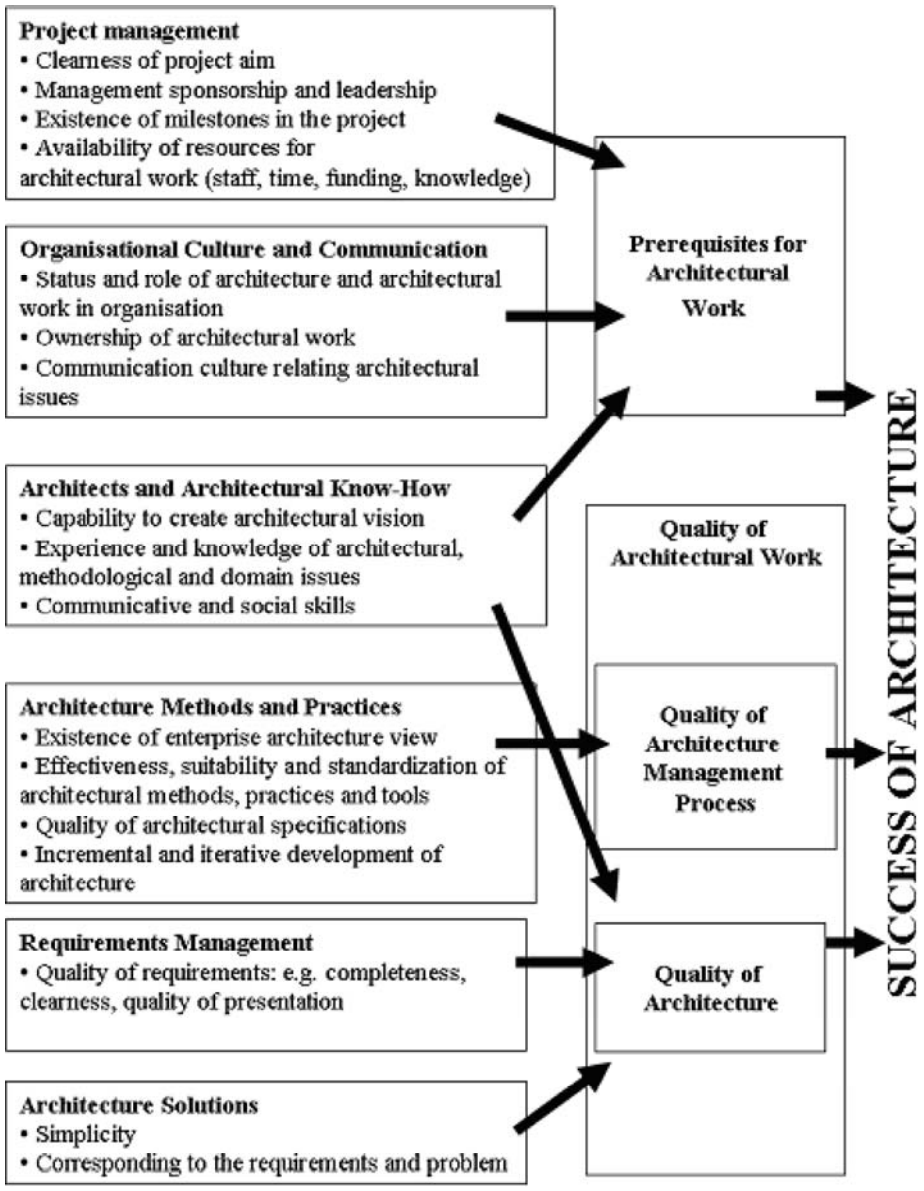


Figure: System development areas affecting the success and failure of software architecture.

The results of this study can be used as a checklist by which practitioners in ICT service providers and user organizations undertaking, or planning to undertake, software architecture efforts can ensure that their software architecture–related efforts are comprehensive and well-implemented. These results can also help to decrease the chance of failure in architecture development.

For more detailed information, see the paper Success and Failure Factors for Software Architecture or the related report.

Table: Examples of the factors related to the areas affecting the success or failure of SA.

Area	Success Factors	Failure Factors
Project management	• Clear aim of the project • Strong management sponsorship	Problems and deficiencies in the project planning

	• Clear milestones in the project • Strong leadership • Clearly defined tems and roles • Available knowlegge / staff • Teamwork	• Not a clear statement of the problem • The project scope too broad • No project, system or testing planning • The lack of clear milestones in the project • No measures of success Problems in the scheduling Problems in the project funding Problems and deficiencies in staffing • Poor leadership • Stakeholders unclear • lack of resources / talent • Lack of quality assurance organization • Lack of requirement team
Organizational culture	• Architecture is woven into the organizational culture • Ownership: Willingness to take ownership of architecture • Approving attitude towards architecture • Training, teambuilding	• Profit-centre and project culture • Quarterly thinking • "Turf" thinking • Organizational politics • Negative attitude towards architecture and architects • Poor communication • Disparity in the perception of the architecture
Architects and Architectural Know-How	• Practical experience • Domain knowledge • System development knowledge • Capability to create architectural vision • Conceptual thinking • Capability to argue rationally • The ability to outline large entities • Communicative and social skills • Project management skills	• Unconvincing leadership by architects • Incapability to create an architectural vision
Architecture Methods and Practices	Architecture Management Process model: • Incremental and iterative development • Validation of requirements	• Focus is on the methods and tools, not on architecture • No architecture selection decision criteria • No change management • No iterative design

	• The evaluation of architecture • Life-cycle thinking in the architectural design Methods, tools and practices: • Suitable and effective methods and tools • Well-defined limits for architects • Clear rules in the architectural decision making • Change management Standardization of architectural practices Architectural specifications: • Clear and understandable architectural specifications Enterprise architecture: • Defined and described EA	• Outputs not identified • Essential architectural views / aspects not documented • Architectural descriptions are at too low a level, are not detailed enough, or are at too high a level • EA is not defined or described, or it is very heterogeneous
Requirements management	• Complete • Agreed • Well-represented	• Incomplete or unclear requirements • Unbalanced set of requirements • Requirements not prioritized • Requirements not documented • Insufficient resources to support a new requirement have been allocated
Architecture solutions	• Simple architecture • Architecture solves problems	• Architecture does not correspond to the requirements • Architectural decisions are based on the wrong interpretation of requirements • Standards and standard components neglected • External structures or exceptions drive the architecture

Top of page

Stakeholders of Architecture Work

The commitment of key stakeholders, such as top management, is crucial to EA success (see e.g. Syntel 2005) and the significance of identifying, involving and managing key stakeholders is also emphasized in other domains, such as SA, information systems (IS), requirements engineering, and management (see e.g. Boehm 1996; IEEE 2000; Mitchell et al. 1997; Pouloudi 1999; Sharp et al. 1999). Stakeholders may have different, even conflicting needs and perspectives (Jonkers et al. 2006; Kaisler et al. 2005; Morganwalp & Sage 2003), which should be identified and utilized in architecture work. For example, communication is essential in architecture work (see e.g. Lankhorst 2005, and thus the key stakeholders and their requirements for architectural content and its representation need to be identified (see e.g. Armour et al. 1999; Lankhorst 2005). Also while planning architecture evaluation, the architecture work stakeholder groups that may need or require evaluation results need to be defined, and potential ways these stakeholder groups will use the information should be discussed and determined (see the paper on Enterprise Architecture Evaluation Components).

Several definitions for a stakeholder have been proposed (see e.g. Mitchell et al. 1997), including the substantially cited one by Freeman (1984):

> "A stakeholder in an organization is (by definition) any group or individual who can affect or is affected by the achievement of the organization's objectives".

However, in this study, the following, even broader definition from the architecture domain was adapted from the IEEE standard 1471-2000:

> Stakeholder is an individual, team, or organization with interests in, or concerns relative to architecture.

Although the standard originally describes a recommended practice for architectural descriptions of software-intensive systems, a system by definition encompasses IT platforms, applications and systems, and even whole enterprises, making the definition appropriate in the architecture context. Stakeholder concerns, on the other hand, are defined in the following way:

> Concerns are interests related to the development of architecture, its use and any other aspects that are important to one or more stakeholders (IEEE 2000, adapted).

In literature, many architecture work stakeholders are proposed. Stakeholders have certain roles (such as Architect or Acquirer), that can essentially be filled with various individuals, teams and organizations (IEEE 2000). At least 13 architecture work stakeholder roles can be identified and from these (with some additional stakeholders from literature), approximately thirty stakeholder individuals, teams and organizations can be derived. Therefore, to gain any control of this myriad of stakeholders a classification scheme based on some of their common characteristics is needed. A few classification models are provided (e.g. Mitchell et al. 1997; Preiss & Wegmann 2001) but none are validated in the architecture context. Therefore, we applied a simple model by Liimatainen and Koskinen (2007). The model originally classifies IS research into three contexts, including the viewpoints of

- IS producers (e.g. software developers),
- facilitators (e.g. information management and ICT maintenance) and
- users.

In the architecture domain, these roles can be defined as follows:

- **Producers** are the stakeholders carrying out architecture planning and development. They differ from facilitators and end-users in the sense that they not usually manage or maintain architecture, or use it for any other purpose than their primary work. However, some stakeholders such as architects can also be involved in management, maintenance, and even use of architecture.
- **Facilitators** are the stakeholders performing architecture governance, management and maintenance. The role also includes stakeholders that sponsor and support architecture work by e.g. providing resources, requirements or ideas. This role differs from producers because it does not directly conduct EA planning or development. Facilitators are not architecture end-users in the sense that their work directly affects architecture.
- **Users** utilize architecture work and its products (e.g. the architecture itself) in their daily work. The difference between the users and the other roles is that the users do not carry out architecture work or directly affect architecture. However, they can be involved in architecture work by e.g. providing business requirements.

A number of common characteristics can be defined for these stakeholder classes:

- **Producers** could be concerned with carrying out architecture planning and development in a way that (to a reasonable extent) satisfies facilitators' and users' requirements for 1) the content, presentation and quality of the work products (e.g. the architecture), and 2) the impacts (e.g. benefits) of the work or architecture.
- **Facilitators** could be concerned with strategic or operational management, maintenance, or sponsorship of architecture or architecture work. In turn, they may require that 1) certain requirements are taken into account in architecture planning and development, and/or 2) certain impacts are realized by architecture or architecture work.
- **Users** could be concerned with receiving architecture work products and/or impacts that satisfy their requirements (e.g. enable or ease their work). In turn, they could be involved in architecture work by e.g. disclosing requirements and feedback.

In spite of these commonalities, some stakeholders may have multiple roles related to architecture and their concerns could also be diverse. In addition, stakeholders could be classified differently depending on the organization and the phase of the EA program, and they could also be organization-specific. Therefore, differences may exist depending on e.g. organizational size, type (e.g. hierarchical or matrix) and industry, and the scope and phase of the EA program. A generic hierarchy or organizational position for an architecture function or governance organization would thus be difficult to define. In organizations worldwide, the architecture function has been commonly situated under CIO or information management, but there seems to be a shift to top business management (Schekkerman 2005).

By successfully identifying and managing architecture work stakeholders, their requirements and other concerns could be more comprehensively and extensively considered in architecture work, potentially resulting in increased organizational satisfaction towards the architecture program. In turn, this may facilitate the diffusion of the architecture approach in the organization. This is especially important because in many organizations, particularly EA is a relatively new discipline. Some stakeholders in organizations may not even be aware that they essentially are architecture work stakeholders.

For more information on architecture work stakeholders, see paper Enterprise

Architecture Stakeholders - a Holistic View.

Top of page

Benefits of Architecture Work

Architecture work, especially on the enterprise level, is claimed to facilitate the realization of a multitude of benefits, such as aligning and integrating strategy, people, business and technology, and enabling organizational agility (see e.g. Goethals et al. 2006; Morganwalp & Sage 2004). However, investments need to be made in organizational, cultural and technical infrastructure to support the architecture program (see e.g. Kaisler et al. 2005) and be justified to the stakeholders by demonstrating the positive effects of architecture work in the organization (see e.g. Morganwalp & Sage 2004).

Nevertheless, presenting the benefits of architecture work is difficult since measuring its effects comprehensively is demanding and the architecture itself is changing constantly (Morganwalp & Sage 2004). This may be one of the reasons why literature provides few academic research results to quantify the argued benefits or value of architecture work, with the exceptions of a few case studies (see Hjort-Madsen 2006; Kamogawa & Okada 2005) and survey-based studies (see Infosys 2005; Ross & Weill 2005; Schekkerman 2005). Still, defining the potential benefits of architecture work is important - it might even be the prerequisite for the selection of objectives for an architecture program, measuring the realized benefits and value of architecture work, and thus providing a rationale for key stakeholder support and investments in architecture (see e.g. Kamogawa & Okada 2005).

On the definition of architecture work benefits, some literature exists and from there, at least 27 individual benefits can be derived, the most referred being

- reduced costs,
- providing a holistic view of the enterprise,
- improved business-IT alignment,
- improved change management,
- improved risk management,
- improved interoperability and integration, and
- shortened cycle times.

The large number of potential benefits suggests that a feasible classification scheme should be applied. However, there does not seem to be an established model for classifying benefits in the architecture context. Therefore, a classification model from the IS domain (see Giaglis et al. 1999) was applied. The horizontal axis of the model distinguishes between quantifiable and non-quantifiable benefits, and the vertical axis between benefits that can be accounted to architecture or architecture work, and those that significantly depend on other organizational or environmental factors as well. In the model, the benefits are categorized into the following categories:

- **Hard bebefits** can be objectively quantified (e.g. in monetary terms, time or other numeric values) and attributed to architecture or architecture work. They could be related to possible cost and cycle time reduction and economies of scale. Moreover, they could include increased standardization attained by utilizing the standards defined in the architecture, increased reuse of architectural models, descriptions and documentation, and increased interoperability between systems constructed according to the architecture. Hence, they can potentially be attributed to

architecture or architecture work.

- **Intangible benefits** cannot be easily quantified, but they can be attributed to architecture or architecture work. These benefits can be realized, particularly, from the development and usage of architectural models and descriptions, leading to better insight of the enterprise and thus supporting e.g. decision making.
- **Indirect benefits** can be measured in quantifiable terms, but cannot be attributed to architecture or architecture work. They are related, especially, to an enterprise's better position in the market, improved management and customer orientation, and more efficient business processes – factors that can be quantified by various metrics but only partially attributed to architecture or architecture work.
- **Strategic benefits** are positive effects that are realized in the long run and are typically affected by a multitude of factors. Therefore, they generally cannot be objectively quantified or completely attributed to architecture or architecture work. These benefits may include, for example, increased stability of an enterprise in an environment of constant change, better strategic agility, and improved alignment with business strategy.

Figure below displays the architectural work benefits derived from literature classified according to the model.

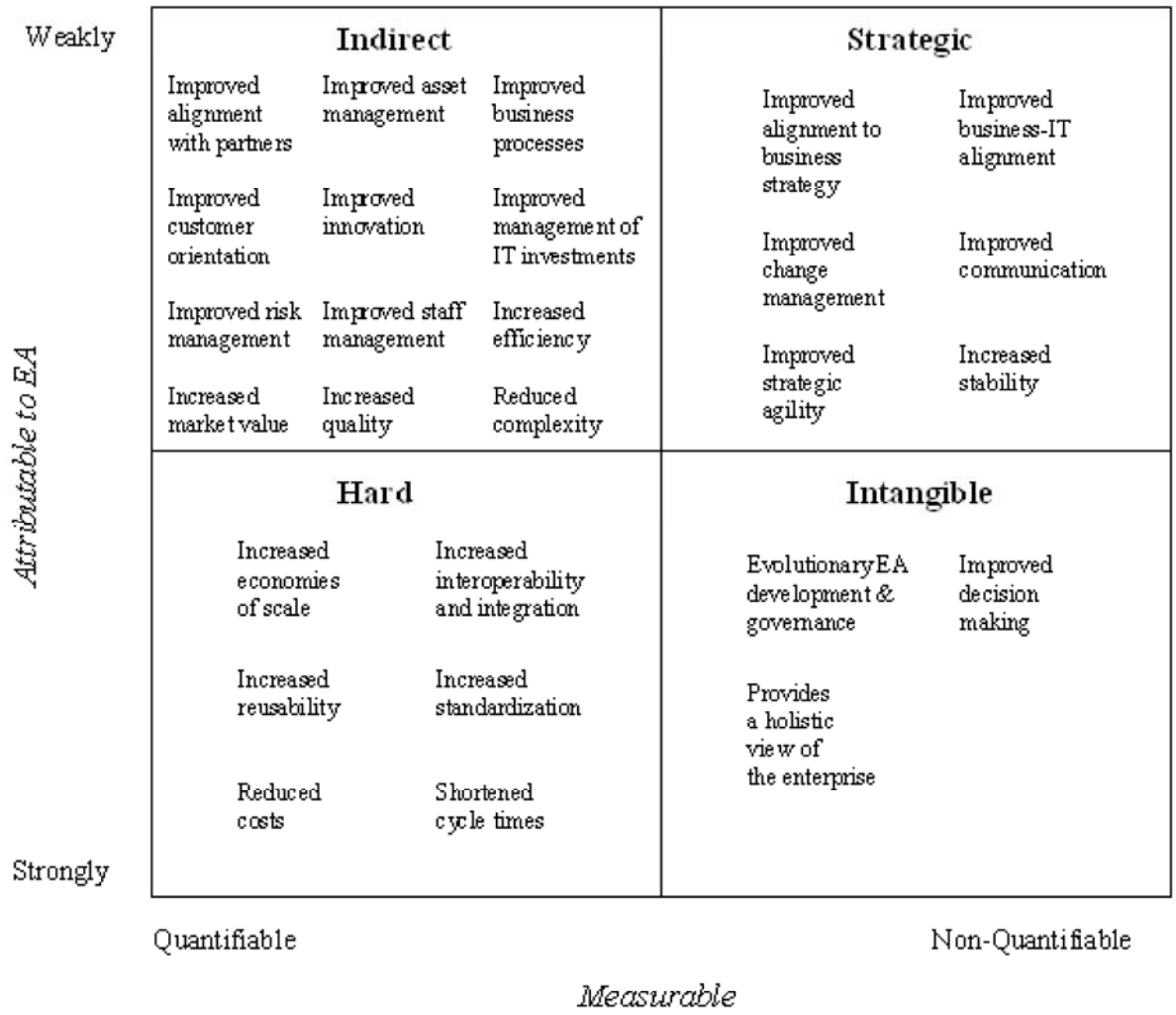


Figure: The classification of the architectural work benefits according to the Giaglis et al. (1999) model.

Judging from the classification, the challenge of evaluating the benefits seems to be that most of them are indirect or strategic - even if they can be clearly quantified, they are difficult to address to architecture or architectural work. Moreover, the relatively large amount of strategic benefits impedes the evaluation as well. Consequently, in the initial stages of architecture maturity, applicable evaluation criteria and metrics for hard benefits could be developed for showing "quick wins". In higher maturity levels however, metrics for other types of benefits should be developed as well to quantify the value of architecture work more comprehensively. Even the indirect and strategic benefits might include elements which could be evaluated and addressed to EA.

At this point, challenges relating to architecture work benefit definition and evaluation still remain. Even though the literature focuses on listing a multitude of potential benefits, it does not clearly define and describe them. Moreover, benefits derived from literature represent different levels of abstraction. Some benefits may also be more like characteristics of architecture or areas of architecture work from which benefits could be gained, even though they are all referred to as benefits in literature. Furthermore, the causes, effects and other relationships between various benefits, architecture characteristics and architecture work activities are not clearly defined, implicating a need for more work on describing the architecture work benefit realization process.

All in all, it is worth noting that architecture work benefits are not automatic. The architecture should be communicated effectively in the organization to realize any benefits (see e.g. Tash 2006). Even then, architecture work does not guarantee long-term value because a multitude of factors affects the realization of benefits (see e.g. Boster et al. 2000). In addition, distinguishing the contribution of architecture or architecture work from all the potential factors affecting the realization of the benefits is clearly a challenge. In any case, architecture and architecture work should be seen as assets, not expenses, and that the expenses are actually realized by not investing in the architecture (see e.g. Tash 2006; Whyte 2005).

For more information on architecture work benefits, see paper Enterprise Architecture Benefits: Perceptions from Literature and Practice. For information on architecture work benefit evaluation, see section Architecture work benefit evaluation and report Evaluating the Benefits of Architectural Work.

Top of page

Status of Architecture Work

During the AISA project the status of architecture work was charted with the help of the Evaluation Model for Enterprise Architecture in the beginning of the year 2006 and again in the beginning of 2008.

The first architecture work status evaluation was conducted in three of the participating companies (see the section on the case studies). Two of the companies represent IT user organizations, in which we studied their internal architecture work status and the third company is an IT service provider, in which we studied its internal architecture work status and the company's view of its customers' architecture work.

In the beginning of the year 2008, the architecture work status was re-evaluated in two case companies representing the IT user organizations.

Data was gathered by semi-structured interviews using the evaluation model to structure

the interview. Each interview took 2-4 hours. In addtion, some internal documents provided by the companies were used to support the analysis and reporting of the interview data. If required, the interview was complemented with an additional phone interview.

The status of each of the architecture work areas in the companies in 2006 can be summarized as follows (see also the figure below displaying the maturity levels of the areas).

1. **Scoping and Purpose:** The cases show that the benefits and objectives of architecture work and the objectives of architecture are mostly identified on a general level. Yet, the identification is somewhat insufficient on occasion.

2. **Organizational Culture:** The general commitment of either or both management and the IT organization already supports architecture work to some degree. A number of organizational challenges for architecture work have been identified and preliminary solutions considered.

3. **Commitment:** The management of the case companies is committed to the architecture approach and is aware of the importance of architecture. In practice however, gaining management support for architecture work is challenging. In all cases, IT organizations are committed to architecture work, but gaining the commitment of business end-users is evidently a challenge. Guidelines for IT developers for assuring architecture compliance are under construction.

4. **Communication & Common Language:** All of the case companies have established communication between the architecture team and the key stakeholders to some extent. Nonetheless, there is room for improvement in communication with management, IT developers and business end-users. A number of architecture concepts are defined, and communication challenges have been discussed.

5. **Development Methodology, Framework and Tool Support:** There are no specific development methodologies for architecture development defined in the case companies. However, methodologies for individual systems and software development projects are mostly well developed and a number of system architecture development methodologies are available. On the other hand, the framework for architecture development is defined and documented in all cases. However, there are challenges either in communicating the framework to all relevant stakeholders or actively using it in architecture development. Multiple tools, including modeling tools, are used in architecture work. Nevertheless, defined and controlled use of the tools is not yet established. Also, the challenges in transferring architectural descriptions between tools are being considered. The use of UML has been discussed, but it is thought to have its limitations in intelligibility.

6. **Architecture Models and Artefacts:** There are a number of deficiencies in the descriptions of the state of architecture, both current and objective, as well as in the transition plans. They are fragmented or based on incomplete information, or even non-existent. Plans for architecture documentation have mostly been done, but the documentation process is typically not very systematic.

7. **Assessment / Evaluation:** The possibilities of architecture evaluation have not been charted extensively or in detail. Nonetheless, evaluations are occasionally made in one or two case companies. A defined set of architecture evaluation methods and metrics is rarely established.

8. **Governance** The organization, functions and processes of architecture governance have been partially defined and documented, but not implemented. Furthermore, the governance unit is mostly situated under IT management. Therefore, the connection between business and architecture has not been fully established. In architecture risk management and organizational change management, existing practices could be used.

9. **Skilled Team and Training / Education:** Most of the case companies have defined roles and responsibilities for a full-time architecture team. The roles of chief architect and business architect are mostly not named, but there is practically a certain person that works in the role of the chief architect. The training and education needs of the team or other stakeholders, such as management, are not yet thoroughly charted. Training and education is available and personal training and education plans are implemented but not actively used.

10. **Project Management:** The case companies have established project management practices. From architecture point of view, the coordination between projects generally operates well. However, methods for collecting project management best practices are not established in all case companies.

11. **Business-Drivenness:** The architecture work of the case companies is mostly driven by business needs and requirements. However, collecting the business requirements and verifying their traceability to e.g. architecture decisions are a challenge.

12. **IT Investment and Acquisition Strategies** It is perceived that investments should be driven by architecture planning and development. A missing link between the investment process and architecture planning is considered as a risk, and therefore the case companies either have connected or are currently establishing a connection between them.

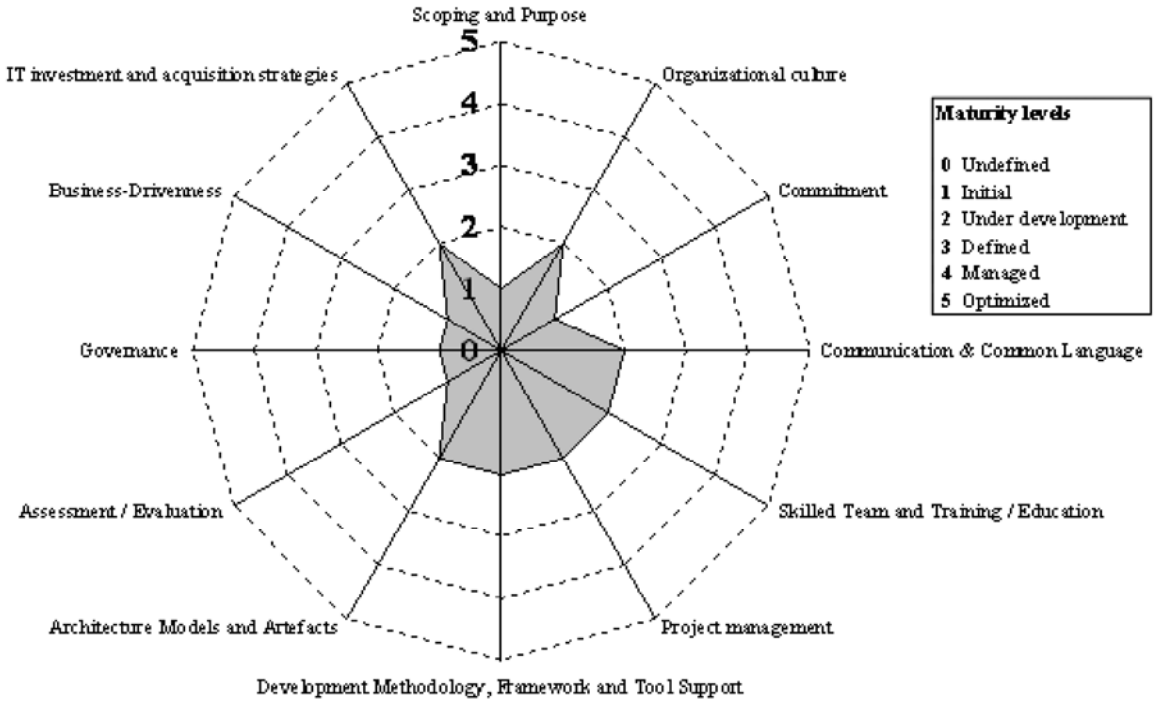


Figure: The average maturity levels of the architecture work areas in the case companies.

The status of architecture work was re-evaluated in the beginning of the year 2008. **The main improvements of the architecture work in the organizations** (compared to the previous status evaluation) relate to the following areas:

- **Architecture framework, development methodology and tools:** Architecture frameworks are better established and are used to guide both the development and management of EA. Actual architecture methods are not necessarily used, but organizations are developing and stabilizing their architecture processes. Architecture toolsets have been further defined.
- **Architecture governance:** Especially the architecture guidance provided to development projects has evolved and proven to be useful in organizations.
- **Teamwork, training and education:** Architecture teams are more stabilized; their roles, tasks, and responsibilities have been clarified. The role of an enterprise architect usually exists in organizations. Architectural skills and capabilities have been acquired e.g. in the form of seminars and lectures. To some extent, architects have also provided training for the in-house stakeholders to increase the commitment and understanding of the architecture work, its objectives and benefits.
- **Business-Driven Approach:** Business linkage has been strengthened; business neeeds and objectives are taken into account more explicitly in the architecture development and enterprise architects are to some extent involved in the organization-level strategy formulation. Also the linkage between investment and acquisition strategies and architecture development has become more evident.
- **Communication and commitment:** Face-to-face communication is considered to be perharps the most effective way to communicate the architecture issues to various stakeholders. The down-ward communication towards the development projects is well taken care of and supports the architecture guidance provided to the projects. The up-ward communication towards the top-management remains a challenge until the architecture work is a stabilized activity in the organizations and some explicit benefits can be shown.

Findings from other studies are mostly parallel with our results. All of the organizations studied by Scherkkerman (2005) have an architecture framework, and virtually all use tools and modeling techniques. Typically, architecture modeling tools include Microsoft Visio and similar programs (Aziz and Obitz 2007). The majority of the organizations also employ architects of their own, but their education and training is most commonly their own responsibility (Schekkerman 2005). Architecture governance is usually located under IT management but there seems to be a shift to business management (Schekkerman 2005). Furthermore, about 60% of the organizations studied have a full-time architecture team (Aziz and Obitz 2007). However, architecture assessment and evaluation seem to be more established in the organizations studied by NASCIO and Infosys (NASCIO 2005; Aziz and Obitz 2007). According to the former, the most of the organizations have or plan to establish architecture performance metrics, while only a third of them do not. According to the latter, about 40 % of the organizations have no architecture metrics.

For more information on the status of architecture work (year 2006), see

- the report Architectural Work Status: Challenges and Developmental Potential - A Case Study of Three Finnish Business Enterprises,
- the paper Architectural Work Status: Challenges and Developmental Potential - A Case Study of Three Finnish Business Enterprises, or
- the paper Enterprise Architecture Work Owerview in Three Finnish Business Enterprises.

Authors: Tanja Ylimäki & Niina Hämäläinen

Quality and Quality Management in Architecture Work

On this page, the following topics are discussed:

- Quality and Quality Management in General: Brief introduction to quality and quality management on a general level
- Quality and Quality Management in EA and SA Context: Introduction to quality and quality management in architecture work

Quality and Quality Management in General

First, we need to define what quality means. Juran & Godfrey (2000) presents the following two meanings for the word 'quality':

- "Quality means those features of a product which meet customer needs and thereby provide customer satisfaction."
- "Quality means freedom from deficiencies - freedom from errors that require doing work over again (rework) or that result in field failures, customer dissatisfaction, customer claims, and so on."

Lecklin (2002) and Dale (2003) describe quality (of a product, service, etc.) for example with the help of the following characteristics:

- conformance to agreed and fully understood requirements,
- fitness for purpose or use, and
- customer satisfaction: the product or service satisfies customer expectations and understands their needs and future requirements in a cost-effective way.

Why we should care about quality in the first place? Dale (2003) presents various points why quality is perceived to be important. Examples of these are as follows:

- quality is a primary buying argument for the ultimate customer,
- quality is a major means of reducing cost,
- quality is a major means for improving flexibility and responsiveness, and
- quality is a major means for reducing throughput time.

How should quality be managed, then? Juran (Juran and Godfrey 2000) introduces his **Trilogy of Quality Management**, which defines that managing for quality makes extensive use of three managerial processes:

1. quality planning,
2. quality control, and
3. quality improvement.

Quality planning can be defined as a

> "structured process for developing products (both goods and services) that ensures that customer needs are met by the final result. The tools and

> methods of quality planning are incorporated along with the technological tools for the particular product being developed and delivered." (Juran and Godfrey 2000)

Quality planning has to deal with the quality gaps depicted in the figure below by providing processes, methods, tools and techniques for closing each of the component gaps and thereby ensuring that the final quality gap is at a minimum.

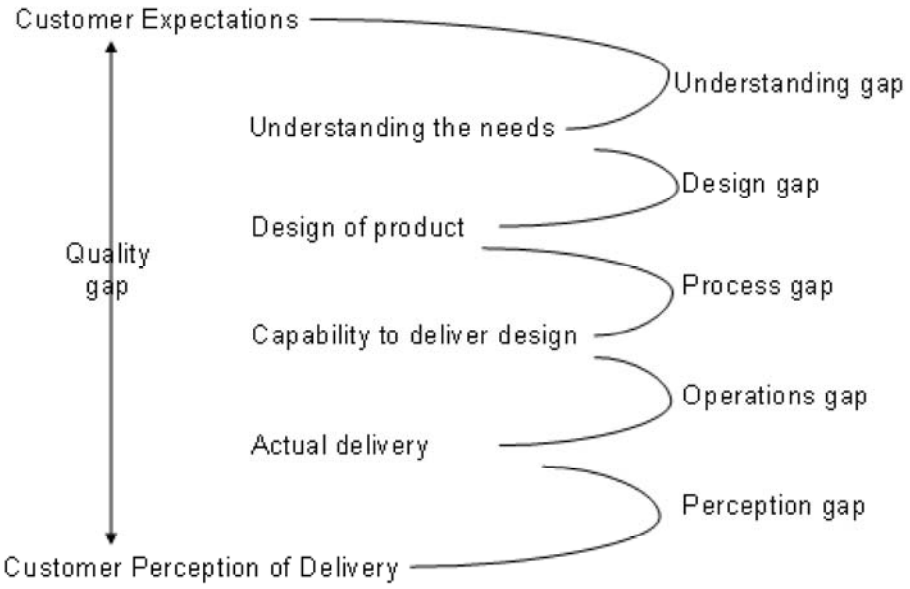


Figure: The quality planning deals with the quality gaps (Juran and Godfrey 2000).

The quality control process is

> "a universal managerial process for conducting operations so as to provide stability - to prevent adverse change and to maintain the status quo." (Juran and Godfrey 2000)

To maintain stability, the quality control process evaluates actual performance, compares actual performance to goals, and takes action on the difference. According Juran quality control's relation to quality assurance can be described as follows:

> "Each evaluates performance, each compares performance to goals, each acts on the difference. However, quality control has as its primary purpose to maintain control (or stability), performance is evaluated during operations. **Quality assurance's** main purpose is to verify that control is being maintained, performance is evaluated after operations."

Quality improvement process is clarified with the definition of the term improvement. It can be seen as an

> "organized creation of beneficial change; the attainment of unprecedented levels of performance." (Juran and Godfrey 2000)

Furthermore, improvement usually takes place project by project and step by step.

Another quality management appoach is **the Total Quality Management (TQM)**. It is a

> "management philosophy embracing all activities through which the needs and expectations of the customer and the community, and the objectives of the organization are satisfied in the most efficient and cost effective way by maximizing the potential of all employees in a continuing drive for improvement" (Dale 1994),

or

> "the vast collection of philosophies, concepts, methods, and tools now being used throughout the world to manage quality." (Juran and Godfrey 2000)

Dale (1994, 21) describes the TQM to evolve through four stages:

- **Inspection**: Activities such as measuring, examining, testing, gauging one or more characteristics of a product or service and comparing these with specified requirements to determine conformity.
- **Quality control**: The operational techniques and activities that are used to fulfill requirements for quality.
- **Quality assurance**: All those planned and systematic actions necessary to provide adequate confidence that a product or service will satisfy given requirements for quality.
- **Total quality management** is the fourth and the highest level and it involves the application of quality management principles to all aspects of the business, including customers and suppliers.

As a conclusion, we want to point out that quality management is not a separate part of the organization, it is more or less integrated into the management system of an organization to enable systematic deployment of the management's strategies and declarations of will throughout the organization (Lecklin 2002). Quality management also includes and deals with the organizational parts, responsibilities, procedures, processes and resources needed to improve quality (Lillrank 1998).

Top of page

Quality and Quality Management in Enterprise and Software Architecture Context

Product and process quality management practices as well as process maturity and capability assessment practices are widely adopted and introduced in ICT industry. These practices include, among others,

- quality standards (e.g. ISO 9000 standards),
- frameworks for assessment the process maturity of an organization or a project (e.g. CMMI, Software Productivity Research (SPR)),
- and quality award programs (e.g. Malcolm Baldrige, European Quality Award).

EA and SA management processes and their quality management are relatively new parts of organisations' processes. Software architecture management (SAM) consists of the activities of capturing the architectural requirements of software-intensive systems and understanding them. Moreover, the process also includes design, analysis/evaluation, implementation, maintenance, improvement, and certification of the architecture as well as its documentation (IEEE, 2000; Bass et al., 1998).

The quality management activities of SA management can be divided into

- **Activities that relate to the quality management of SAM process:** These activities concentrate on the quality of SAM-process (process quality aspect).
- **Activities that relate to the quality management of SA:** These activities concentrate on the achievement of software architecture of good quality (product quality aspect).

Similary, **the quality management of EA** was addressed. An EA, to be successful, needs to be understood, accepted and used in everyday business functions, including also the various activities conducted by the top-management. The success needs also to be measured in order to ensure that desired results are achieved. While there is no widely accepted definition of a high-quality EA, we have suggested (see the papers Potential Critical Success Factors for Enterprise Architecture and Towards a Generic Evaluation Model for Enterprise Architecture) that **EA has high quality** if it

- conforms to the agreed and fully understood business requirements,
- fits for the purpose, which is to gain business value through EA, and
- satisfies the different stakeholders' (e.g. the top management, IT management, architects, developers) expectations in a cost-effective way and understands their current needs as well as the future requirements.

Briefly, different stakeholders profit from the high-quality architecture work and its results. Especially, EA should provide the management a clear view of the top priority projects the organization needs to carry out in the first place. Furthermore, the different views of EA quality presented above implicitly imply that the quality of EA is more than merely the quality of the implemented EA, indicating that it is successfully used. The quality of EA may also refer to the quality of EA documentation, the quality of the EA development process, the quality of EA governance (process), and so forth.

Quality management (QM) of EA is about defining and conducting all those activities that are needed to reach an EA of high quality and, thus, it relates to the same perspectives than the quality of EA. There is a need to manage e.g. the quality of EA governance process, EA development process, EA artifacts or specification, and the implemented EA that is used. We suggest that, as presented in the figure below, **QM activities for EA** are integrated into

- the EA governance process and
- the EA development life cycle.

Quality management of the EA artifacts is included in the QM activities that are integrated into the EA development life cycle.

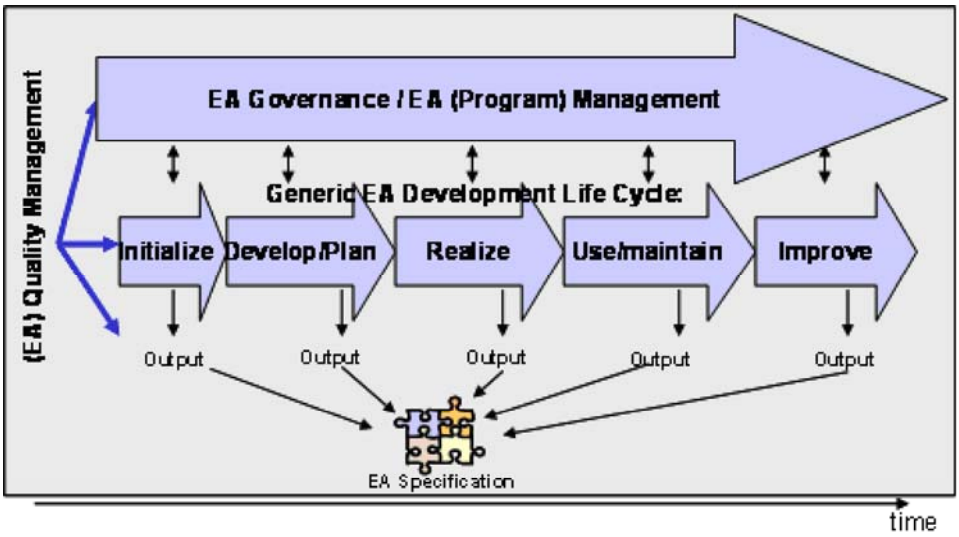


Figure: EA quality management is integrated into the EA governance process and the EA development life cycle.

As a conclusion, we state that architecture management, both on EA and SA level, is spread out to many processes in organisations and there is a need

- to move from architectures driven by investment planning and system development towards architectures driven by architecture management
- of architecture management practices and process models that aim at high-quality architectures
- to advance the maturity of architecture management processes
- for agility in architecture management and development
- for metrics and metric programs for architectural maturity and quality.

More detailed information on quality management of SA can be found in

- the paper Quality Management Activities for Software Architecture and Software Architecture Process or
- the report Quality Management Activities in Software Architecture Process.

Similarly, more detailed information on the quality management of EA can be found in the report Quality Management Activities for Enterprise Architecture.

Top of page Page Index

Authors: Niina Hämäläinen, Tanja Ylimäki and Eetu Niemi

Evaluating Architecture Work

On this page, the following topics are addressed:

- Overview - Measurement and Evaluation in General: Introduction to measurement and evaluation issues on a general level
- Needs and Triggers for Measurement and Evaluation: Description of measurement and evaluation needs and drivers in architecture work
- Evaluation Aspects in Architecture Work: Description of the various aspects of measurement and evaluation in architecture work
- Status of Measurement and Evaluation in Architecture Work: Description of the status of measurement and evaluation in organizations' architecture work

Overview - Measurement and Evaluation in General

Evaluation and measurement pervades almost every facet of our lives and daily activities. Much of what we do, decisions we make, and decisions made about us involve measurement or evaluation of one kind or another. A discipline of evaluation is needed because companies and societies in general require systematic, unbiased means of knowing if their products, processes, programs, and personnel are good (Shadish et al., 1991).

Evaluation and measurement concepts and practices are developed in different domains, such as in

- program management,
- software engineering, or
- quality management.

Evaluation and measurement may therefore mean different things to different people. **Evaluation** can be defined as

> "the process of determining the merit, worth, and value of things, and evaluations are the products of that process" (Scriven, 1991)

or as

> "a study designed and conducted to assist some audience to assess an object's merit and worth" (Stufflebeam, 2001).

Evaluation can also be seen as a means to generate information that assists in making judgments and decisions, for example about a program, service, policy, organization, or person (Stufflebeam, 2001).

Measurement can be defined as the assignment of numbers to aspects of objects or events to one or another rule or convention (Stevens, 1968) and as the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules (Fenton, 1994).

Evaluation and measurement are carried out for many different purposes in companies;

for example to (Behn 2003)

- **evaluate**: how well is the organization/unit/team/people performing?
- **control**: how to ensure that the subordinates are doing the right thing?
- **budget**: on what programs, people or projects should resources be allocated?
- **motivate**: how to motivate e.g. line staff, middle managers, and stakeholders?
- **promote**: marketing/public relations aspect; how to convince stakeholders that the organization/unit/team is doing a good job?
- **celebrate**: what accomplishments are worthy of the important organizational ritual of celebrating success?
- **learn**: why is what working or not working?
- **improve**: what exactly should who do differently to improve performance?

More information on the issue can be found in the dissertation Evaluation and Measurement in Enterprise and Software Architecture Management.

Top of page

Needs and Triggers for Measurement and Evaluation

Architecture evaluation is a way to get answers to organization's information needs and problems relating to its business and ICT. Some of the reasons for the increasing interest in the evaluations and measurements of architectures are the facts that companies have needs to move towards business value driven ICT-development and there are pressures to improve the cost-effectiveness of ICT. However, the role and the meaning which architecture evaluation may have in companies are not clearly identified or defined.

In the AISA project, a study was conducted to gain an understanding of the roles and meanings of architecture evaluation and measurement in business organizations. Triggers for evaluations and measurements were identified and analyzed. The study revealed that architecture evaluation may

- enhance the understanding of organization's business and ICT-environments from financial and structural viewpoints, or
- be used as a tool in change management, quality assurance, process planning, IT cost management and architectural choice making.

It seems to be difficult for the practitioners to directly specify evaluation needs that relate to each architectural view (views are described in more detail in the article The Role of Architecture Evaluations in ICT-companies). It was suggested that organization's business and ICT-related problems, questions, topics of concern and information needs may be triggers for architecture evaluations.

The triggers can be categorised to the following categories:

- **Company and business management**: Needs to support organization's structural design (e.g. business process design) and distribution of the work (e.g for out-sourcing).
- **Holistic view**: Needs to understand the current status of organization's business and ICT-environment.
- **IT cost management**: Financial information needs relating to ICT (applications and technical infrastructure).
- **Change management**: Change pressures relating to architectures and

architectural principles - identification of probability and nature of changes that should be made and decision making about changes.
- **Quality management**: Quality questions relating architectural documentation, the information/data structures, application and technical infrastructure, as well as systems solutions.
- **Architecture management**: Confirming that architecture related work meets expectations e.g. investments correspond to the architectural principles.
- **Architectural choices**: evaluation of architectural alternatives against quality, cost and other aspects.

We suggest that **these evaluation triggers describe the role and meaning that architecture evaluation** may have in companies.

For more information on the subject, such as examples of triggers which came up in the study and the evaluation needs which arise due to these triggers, see article The Role of Architecture Evaluations in ICT-companies.

Top of page

Evaluation Aspects in Architecture Work

In the AISA project, a study was conducted to support the planning of metrics for EA programs by presenting measurement aspects and phases of iterative and goal-oriented metrics development process. A Goal Question Metrics (GQM) (Basili, et al. 1994) approach is perhaps the most well-known goal-oriented approach to measurement planning.

The approach is based on the assumption that for an organization to measure in a purposeful way it must

- specify the goals for itself and for its projects,
- trace those goals to the data that are intended to define goals operationally, and finally
- provide a framework for interpreting data with respect to the stated goals (Basili, et al. 1994).

It is, therefore, important to clarify, at least in general terms, what information needs the organization has, so that these needs can be quantified whenever possible, and the quantified information can be analyzed to whether or not the goals are achieved (Basili, et al. 1994). GQM-approach uses a top-down approach to define metrics and a bottom-up approach for analysis and interpretation of measurement data (Ardimento, et al. 2004). GQM is highly iterative process, e.g. goals are identified during working with questions (Berander and Jönsson 2006).

In the study, the following **measurement aspects for EA program** were identified:

- Benefits of EA program for organization
- Impacts and use of EA program and its results
- Progress and operations of EA program: EA team's and architects' accomplishments, particularly the progress towards the pre-established goals
    - Quality / maturity
    - Maturity of EA program capabilities

    - Quality of results produced by EA program

- Architecture structures in organization: evaluation of architecture alternatives and solutions

These aspects can be used to support the identification of company's measurement needs and derivation of related metrics.

More detailed information on the subject, especially the phases of iterative and goal-oriented metrics development process, can be found in paper A Goal-Oriented Way to Define Metrics for Enterprise Architecture Program.

Top of page

Status of Measurement and Evaluation in Architecture Work

In this section, the current status of measurement and evaluation in both the EA and the SA domains are briefly addressed.

Views about EA evaluation meanings seem not to be yet stabilized. However, roughly categorizing, evaluation approaches and techniques in EA management domain seem to relate to the following two aspects:

- Approaches and techniques that generate information relating to the company's EA program and its results (e.g. EA program's efficiency, effectiveness, maturity, quality of results) to support planning, improvement, marketing (showing value), organization and management of enterprise architecture work in a company. Company's business and IT goals are quite common used as the starting point in these evaluations.
- Approaches and techniques that generate information to support decision-making on the enterprise-wide information system issues through the analysis the EA models. This aspect can also be referred to as property oriented enterprise architecture evaluation.

A variety of metrics, evaluation criteria and methods have been developed for the evaluation of EA programs and their results. These include

- generic process metrics for evaluating EA activities,
- architectural documentation criteria for evaluating the results of EA programs,
- EA maturity models,
- EA acceptance and use metrics, and
- EA benefit and value measures.

However, generic, validated measures for EA acceptance or benefits have not been presented thus far and therefore have to be developed according to the specific requirements of individual companies. On the other hand, the most typical published EA evaluation methods include EA maturity models that measure the overall 'EA capability' of the organization, i.e. the capability of the organization to manage the development, implementation and maintenance of its EA (see e.g. the paper Towards a Generic Evaluation Model for Enterprise Architecture). Some of the maturity models also include the realized benefits of EA or the quality of architecture documentation as evaluation criteria. EA maturity models include, for example

- OMB Enterprise Architecture Assessment Framework (OMB, 2005)
- The Enterprise Architecture Maturity Model, EAMM (NASCIO, 2003)
- The Extended Enterprise Architecture Maturity Model, E2AMM (Schekkerman, 2003)
- A Framework for Assessing and Improving Enterprise Architecture Management, EAMFF (GAO, 2003)
- The COSM (Component Oriented Software Manufacturing) Maturity Model (Herzum Software)
- IT Architecture Capability Maturity Model, ACMM (DoC, 2003).

Similarly, a variety of methods and techniques have been developed to support the decision-making on the enterprise-wide information system issues. These techniques are also called as **property oriented EA evaluation techniques** and they can be categorized as follows (Winter et al., 2007):

- Dependency analysis exploits the associations between the various EA artifacts to derive direct and indirect dependencies between these artifacts. A typical analysis question might be 'Which business processes are affected if we switch-off a certain server?'
- Coverage analysis usually spawns two or more EA layers. The results of this analysis technique are often represented as matrices relating the two dimensions of interest.
- Interface analysis focuses on the interfaces within a class of EA artifacts. A typical example is the analysis of technical interfaces between software components specified within software architecture.
- Heterogeneity analysis tries to identify those architecture elements which should be reconsidered and re-factored to improve overall architecture homogeneity.
- Complexity analysis is strongly related to interface analysis. The design goal is to reduce the overall EA complexity.
- Compliance analysis aims to check whether certain policies (like process and data ownership) are defined at a certain organizational level of absraction or if certain mechanisms (like authorization and recovery) have been implemented at a certain software system level of abstraction.
- Cost analysis calculates and reports the costs induced by creation and maintenance of various EA artifacts (e.g. the cost for launching a new product). An important application of cost analysis techniques is the calculation of IT-related costs and the allocation of these costs to products, services, processes, organizational units, etc.
- Benefit analysis is complementary to cost analysis. It exhibits the contributions of individual organizational units, products, application systems and similar artifacts to the overall goals of the organization.

A formal **SA evaluation** is seen as an essential standard part of the architecture-based software development life cycle (SEI, 2007). Companies are now adopting architecture evaluations as part of their standard software engineering development practice, and some are including these evaluations as part of their contracting language when dealing with subcontractors (Kazman & Bass, 2002). The software architecture evaluation is designed to answer to the question 'Will the information system to be built from this architecture satisfy its business goals?' (Kazman & Bass, 2002) Furthermore, the purpose of evaluation is

> to determine the quality of an architectural description and to predict the quality of systems whose architectures conform to the architectural description (ISO, 2007, IEEE 2000).

Software architecture evaluations are seen valuable because they (Maranzano et al., 2005)

- uncover design problems early in the development when they are not expensive to fix,
- leverage experienced people by using their expertise and experience to help other projects in the company,
- let the companies better manage software components suppliers and provide management with better visibility into technical and project management issues
- generate problem descriptions by having the evaluation team criticize the descriptions for consistency and completeness,
- rapidly identify knowledge gaps and establish training in areas where errors frequently occur,
- promote cross-product knowledge and learning and keep experts engaged, and
- spread knowledge of proven practices in the company.

Evaluation in the SA domain has been studied a lot. Studies have focused for example on the evaluation of SA management (Bass & John, 2001; Kazman & Bass, 2002; Lee & Choi, 2005) and on the metrics and measurement of SA (Chastek & Ferguson, 2006; Dias et al., 1999; Shereshevsky et al., 2001; Tvedt et al., 2002). In addition, an array of methods is also being developed for evaluation of software architectures. Examples of these are

- Scenario-based Architecture Analysis Method, SAAM (Kazman et al., 1994)
- Architecture Trade-off Analysis Method, ATAM (Kazman et al., 1998)
- Active Reviews for Intermediate Design, ARID (Clements, 2000)
- SAAM for Evolution and Reusability (Lung et al., 1997)
- Architecture-Level Modifiability Analysis, ALMA (Bengtsson et al., 2004)
- MITRE's Architecture Quality Assessment (Hilliard et al., 1996; Hilliard et al., 1997).

In addition, different kinds of checklists are developed to evaluate architecture and its description during designing it (e.g. Rozanski & Woods, 2005). In summary, SA evaluation focuses mainly on the quality of architecture itself, not on the quality of architecture design or planning process like the EA evaluation.

More information on the issue can be found in the dissertation Evaluation and Measurement in Enterprise and Software Architecture Management. See also paper The Role of Architecture Evaluations in ICT-companies.

Top of page Page Index

Authors: Eetu Niemi, Tanja Ylimäki & Martin Hoffmann

Evaluation Practices in Architecture Work

On this page, the following topics are addressed:

- Evaluation Planning: Overview of the evaluation planning components
- Generic Evaluation Model for Enterprise Architecture: Description of a model to evaluate the status (the maturity) of an organization's architecture work
- Existing Architecture Evaluation Methods: Description of the existing architecture evaluation methods

Evaluation Planning

EA evaluation literature focuses particularly on defining EA metrics and evaluation criteria, especially in the form of maturity models (see e.g. GAO 2003; IAC 2005; OMB 2005), but almost omitting the aspect of elaborate evaluation planning. However, as EA is extensive and can be approached from a number of viewpoints, EA evaluations need to be planned systematically and require taking into account a broader set of aspects than merely selecting and implementing metrics. Therefore, EA evaluation was approached from the program evaluation perspective and established literature (see Chen 2004; Fitzpatrick et al. 2003; Grasso 2003; Lopez 2000; Shadish et al. 1991; Stufflebeam 2001; Taylor-Powell et al. 1996) and a focus group interview of practitioners was used to define the building blocks - or components - that need to be addressed in EA evaluation planning. The components of EA evaluation are defined as follows:

- **EA Objectives:** The goals set for the EA approach in the organization.
- **Evaluation Purpose:** The reasons for the evaluation to be conducted.
- **Evaluation Target:** The object under evaluation (to delimit the factors to be considered).
- **Evaluation Audience:** The users of the evaluation information and results.
- **Quality Attributes and Metrics:** The characteristics of the target that are to be evaluated.
- **Yardstick or Standard:** The ideal result against which the real result is to be compared.
- **Data Gathering Techniques:** The techniques needed to obtain data to analyze each characteristics of an evaluation target.
- **Data Synthesis Techniques:** Techniques used to judge each characteristic of an evaluation target and, in general, to judge the target, obtaining the results of evaluation.
- **Evaluation Process:** Series of activities and tasks by means of which an evaluation is actually performed.
- **Evaluation Management:** Issues related to responsibilities, resources required (people, budget, timeliness, and so forth) and risks.

Figure below displays the components of EA evaluation. A number of potential relationships between them are depicted as well. The definition of evaluation purposes needs to start with answering the question 'why is the program carried out'. In the context of EA, this requires an understanding of EA objectives; what are the organization's goals of EA and EA work. EA objectives provide a valuable input to EA evaluation planning affecting both the purposes and the targets of EA evaluation, and can thus be regarded as an additional component to be taken into consideration.

Moreover, the evaluation purposes and targets are interrelated with each other. Evaluation audiences, on the other hand, have various evaluation needs and concerns, and thus affect both the evaluation purposes and targets.

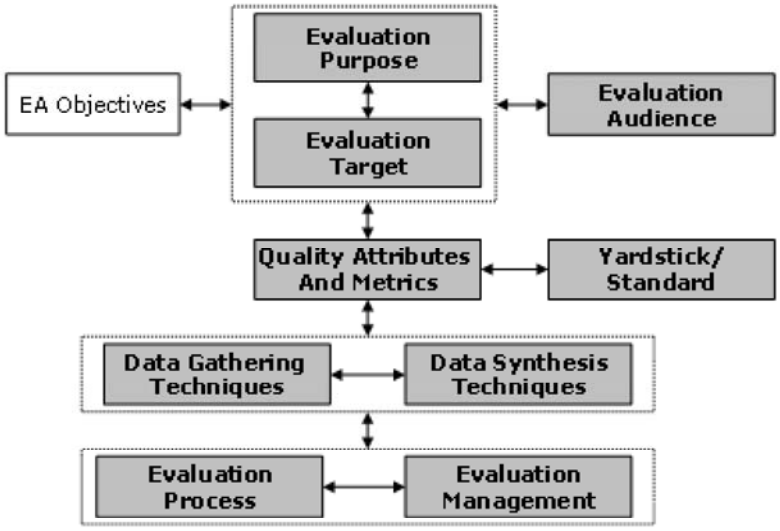


Figure: The components of EA evaluation.

It should be noted that the maturity of the organization's EA affects the selection of evaluation targets, as well as the definition of evaluation criteria and metrics. Thus, the EA maturity level of the organization, the evaluation targets, and the evaluation criteria and metrics need to be compatible. Organizations on lower levels of maturity should start with simple metrics (such as on/off-metrics or quantitative metrics). While the organization matures, more detailed business impacts can potentially be measured. In addtition, it should be considered that the EA evaluation targets and metrics must be compatible with the other evaluation and measurement systems used in the organization (such as Balanced Score Cards).

For more information on EA evaluation planning, see paper Enterprise Architecture Evaluation Components.

Top of page

Generic Evaluation Model for Enterprise Architecture

Several maturity models, which have their origins in the field of quality management (Chrissis et al., 2003; Fraser et al., 2002), have been developed to assess the stage of an organization's EA and to enhance its quality, such as the models developed by the Chief Information Officers Council (1999), the U.S. Department of Commerce (2003), and the Office of Management and Budget (2005).

The maturity of EA refers to an organization's capability of managing the development, implementation and maintenance of its architecture (van der Raadt, et al., 2004), which usually consists of four viewpoints: business, information, systems, and technical architecture (e.g. The Open Group, 2006). Furthermore, the idea of these maturity models is that maturity evolves over time from one level to a more advanced level, without skipping any level in between, eventually moving towards the ideal ultimate state (Klimko, 2001).

In the AISA Project, we regarded these maturity models as one means of advancing the quality of EA by providing at least an initial EA quality management system (see also Cullen, 2006). Something that we considered to be a downside with these maturity models was the fact that they seem to be more or less domain specific; especially developed for the various areas of the public administration (e.g. DoC, 2003; IAC, 2005; NASCIO, 2003). Publicly available maturity models, specifically suitable for evaluating the EA of heterogeneous private sector companies, are still hard to find. Hence, we decided to take another approach to the problem: we applied the concept of a Critical Success Factor (CSF) to the field of EA and defined the potential CSFs for EA. These CSFs represent the factors that have to be carried out exceedingly well in order to attain successful EA, a high-quality EA.

Furthermore, we aimed at developing a generic evaluation model for Enterprise Architecture (later the model is referred to as gemEA), a model that is suitable for evaluating the stages of EA in private sector organizations, regardless of their line of business. Consequently, **the initial gemEA** consists of the following three parts:

1. the set of 12 potential CSFs for EA representing the areas to be evaluated,
2. the key questions assigned to each CSF, and
3. the maturity levels to evaluate the stage of each CSF.

The maturity levels, shown in the table below, were derived from the existing maturity models (Chrissis et al., 2003; DoC, 2003; GAO, 2003;NASCIO, 2003; OMB, 2005). The aim was to define the maturity levels in such a way that they can be used for evaluating the stage of all the diverse areas (the CSFs) in the gemEA.

Level	Level Name	Description
0	Undefined /None	No evidence of any kind of the particular area being taken into account.
1	Initial	The need for taking the particular area into account has been recognized. Artifacts and practices may exist, but they may be incomplete or inconsistent. Processes are mainly informal and ad-hoc.
2	Under Development	Artifacts and documented practices or processes exist. Some may be even complete. Implementation or deployment is not yet carried out. Practices or processes are not yet utilized.
3	Defined	Practices or processes and artifacts have been completed, accepted and communicated to the stakeholders. Implementation, deployment, and utilization have started.
4	Managed and Measured	Implemented or deployed. Practices or processes and artifacts are being utilized and considered as part of normal operations in the organization. Practices or processes and artifacts etc. are measured against a set of predefined and established metrics or criteria.
5	Optimizing (continuous improvement)	Practices or processes related to the particular area are continuously improved. More specifically, clear proofs of architecture benefits, e.g. demonstrable improvements in efficiency, cost savings and service quality, can be seen.

The initial gemEA was tested in the three organizations participating in the research project (see the section describing the case studies). Based on the trial use of the gemEA, it seems that the model briefly described above is suitable for evaluating the current stage of EA in various types of private sector organizations (representing IT user organizations). Furthermore, the gemEA provides a tool to evaluate an IT service-provider organization's ability to deliver EA development and management services and practices for its customers.

The CSFs in the gemEA take various viewpoints into account and provide a more holistic and extensive view to an organization's EA than most of the existing models. In addition, the gemEA is also generic enough to enable the evaluation of the state of EA in various organizations representing different lines of businesses; whereas, most existing maturity models that have been used in the EA evaluation are defined in terms of public sector organizations only.

The main improvement needs detected are the following:

- **Categorization of the questions attached to each CSF in the gemEA**: Two or three levels of questions for each CSF could be determined; general-level questions supported by more detailed questions. This categorization would make the application of the model more flexible.
- **Prioritization or weighting either 1) the CSFs, 2) the different parts of the CSFs, or 3) both**: During the analysis of the interview data, it was noted that difficulties may appear in assessing the maturity of a CSF if it consists of several different aspects; which part of a CSF should be emphasized and why? One solution to this problem would be the prioritization of the CSFs, or perhaps the weighting of them, as well as the different aspects within a CSF
- **Combining or dividing the CSFs in the gemEA**: Depending on the organization's needs (or the phase of the EA development), there may be a need to divide some CSFs into several separate parts (such as framework, development methodology and tool support), especially if there seems to be a lot of variation in the maturity or development activity among these parts
- **Organization of the CSFs**: Should the CSFs be organized or categorized further? How should they be categorized? One possible grouping for the CSFs was found, namely:

    1. **Architectural starting points** including Scoping and Purpose; Organizational Culture; Commitment; Communication and Common Language
    2. **Methods and tools for architecture work** including Development Methodology and Tool Support; EA Models and Artifacts; Assessment and Evaluation
    3. **Support for architecture work** including Governance; Skilled Team, Training and Education; Project and Program Management
    4. **Integration with the organization's other processes** including Business Driven Approach; IT Investment and Acquisition Strategies

    This categorization provides one possible way of interpreting the results. For example, it may help in depicting the extent to which the organization has addressed the architectural starting points, which are crucial in facilitating the further EA development.

More information on the evaluation model can be found in the following articles:

- Potential CSFs for EA desribing the CSFs and the questions related to these CSFs.

See also the related report
- Towards a Generic Evaluation Model for Enterprise Architecture describing the evaluation instrument and the results of its trial use. See also the section Current status of architecture work.

Top of page

Existing Architecture Evaluation Methods

The evaluation of EA regarding its quality and benefits is rather difficult even though a large number of architecture evaluation methods have been introduced. In the AISA project, a research was conducted to review these existing methods. Methods were classified according to the general views of EA; business, information, software systems and technology architectures.

The methods suggested for the business architecture are

- governance modelling (e.g. Yu & Deng, 2006),
- business process modelling and simulation (e.g. Vidovic, 2003), and
- financial methods for assessing the value of an IT investment (prediction of expected benefits through IT investment) (e.g. Symons, 2006).

The needs concerning the enterprise's information architecture were addressed by the evaluation of the corporate data model which is a structured conceptual model of the organisation's data entities and their relations (see e.g. Goodhue, Wybo et al., 1994). The suggested methodology was the Moody's Framework (Moody, Shanks et al., 1998).

The systems architecture consists of software systems. A software system is described through software architectural artefacts. Therefore, the evaluation techniques suggested for the systems architecture are methods for software architecture evaluation

- questionnaires (Bass, Clements et al., 2003),
- scenario-based methods (Clements, Kazman et al., 2002),
- design metrics (see e.g. Clements, Kazman et al., 2002),
- prototyping (e.g. Mårtensson, Grahn et al., 2003), and
- mathematical modelling (e.g. Bosch and Molin, 1999).

Since the infrastructure which allows the deployment of software applications is also part of the software system the underlying execution environment can be evaluated within the software architecture evaluation. The methods concerning the software system evaluation enable predictions regarding the whole system life cycle. Especially, characteristics, such as performance, cost, reliability and maintenance are essential characteristics in the enterprise architecture context. The suggested methods, such as '4+1 Model of Architecture' (Kruchten 1995), are able to assess these criteria.

All reviewed methods, standards, and measures address EA related concerns and evaluation needs regarding business, information, systems, and technology. All of the reviewed techniques have been developed or tested and validated in a practical environment. Many of the introduced evaluation methods rely on conceptual models which improve the architectural awareness and knowledge sharing among stakeholders from different domains. As possibilities to evaluate the ICT architecture, SA evaluation methods and benchmarking are given. Furthermore, financial methods for assessing the

business value of ICT investments are presented. These methods result in the financial measures costs and benefits of ICT related investment decisions. The measures are adjusted to risks and possible change influences.

For more detailed information on existing architecture evaluation methods, see paper Analysis of the current State of Enterprise Architecture Evaluation Methods and Practices and report Architecture Evaluation Methods.

Top of page Page Index

Authors: Eetu Niemi, Niina Hämäläinen and Tanja Ylimäki

Metrics and Criteria for Evaluating Architecture Work

On this page, metrics and criteria for the following evaluation targets are addressed:

- Architecture Documentation: Overview of evaluating the quality of architecture documentation
- Communication: Overview of evaluating the architecture related communication
- Commitment: Overview of evaluating the commitment to the architecture work
- EA Compliance: Overview of the evaluating the EA compliance
- Business-IT Alignment: Overview of evaluating the business-IT alignment
- Benefits of Architecture Work: Overview of evaluating the benefits of architecture work

Architecture Documentation

Architecture documents (architecture descriptions) have a more and more central role in the company management, IT governance and system development. Models are essential elements of architectural descriptions (AD) (Rozanski & Woods 2005). Models act as a medium for communication, helping to explain thinking to others. Models reduce the amount of information the reader needs to understand, and their structure guides the reader through the information (Rozanski & Woods 2005). In addition, models help to understand the situations they are describing and to analyze these situations by allowing to isolate the key elements and understand their relationships. Models also help to organize processes, teams, and deliverables as a result of the structures they reveal in the situation being modeled (Rozanski & Woods 2005).

Architectural documentation is most typically used in business planning for transition from a legacy business or ICT structure to a new structure and in communication, for example between acquirers and developers as a part of contract negotiations (see e.g. IEEE 2000). The quality of architectural documents is crucial for the value of documents: how useful those are for the company's business and ICT development work.

Previous studies have dealt with the quality evaluation of conceptual models (Lindland, Sindre & Solvberg 1994; Claxton & McDougal 2000; Bolloju & Leung 2006) and technical documentation (Smart 2002; Hargis, Carey et al. 2004). Quality dimensions for conceptual models (syntactic, semantic and pragmatic quality) (Lindland, Sindre & Solvberg 1994; Bolloju & Leung 2006) and for technical information (easy to use, to understand and to find) (Hargis, Carey et al. 2004) have also been defined. In addition, quality properties for conceptual models (Lindland, Sindre & Solvberg 1994) and for technical information (Hargis, Carey et al. 2004) are also defined.

In addition, some studies, books and guidelines address, for example, EA descriptions (e.g. Lankhorst 2005; Polikoff & Coyne 2005; Bernus 2003) and SA descriptions (e.g. Rozanski & Woods 2005; Clements, Bachman et al. 2002; Fairbanks 2003; Fu, Dong & He 2002). Qualities of an effective architectural description (e.g. correctness, sufficiency, conciseness, clarity, currency and precision) are also introduced, for example, by Rozanski and Woods (2005).

In the AISA Project, we contributed to the quality assessment of architectural

documentation by identifying and defining a group of questions, criteria and metrics that can be used in the quality assessment of architectural documentation and models. The results of the study aim to help enterprise and software architects to produce architectural descriptions and models of good quality.

Quality of architectural descriptions can be evaluated from the following aspects (see also the figure below):

- **Stakeholder and purpose orientation**: evaluation of how well documents are focused on purpose and on the stakeholder that use these documents.
- **Quality of content**: evaluation of quality of information included in the models.
- **Presentation/visualization quality**: evaluation of how well information is presented in documents.

Figure: Aspects on quality of architecture description.

A set of evaluation criteria and questions to be used for the evaluation of each of these aspects was identified. Furthermore, a set of evaluation factors for the management of architecture documentation was identified. Examples of the evaluation criteria and questions are presented in the table below.

The identified evaluation criteria and questions for architectural documentation can be used by the enterprise and software architects in their architecture design and documentation work, as well as by the reviewers in reviews of architectural documentation.

More information about the evaluation of architectural documentation can be found in report Quality Evaluation of Architectural Documentation and Models and in the paper Quality Evaluation Question Framework for Assessing the Quality of Architecture Documentation.

Table: Examples of evaluation criteria and questions for architecture documentation and descriptions.

Aspect	Criteria	Evaluation Questions/Metrics
Stakeholder and purpose orientation	Stakeholders	• Are the stakeholders of a model/AD defined and who are them?
	Purpose	• Is the purpose of a model/AD in relation to these stakeholders defined and what it the relation?
	Model's/AD's suitability for the	• Does the model provide the stakeholder with the desired knowledge? • Is a practical reason for the information

	stakeholders	evident? • Is the information presented from the stakeholders' point of view?
	The use of AD/models - value of AD/models	• Frequency of use • Number of users • Variety of users (the variety of different functional areas or skill levels of personnel who will likely use this documentation) • Impact of non-use
Quality of content	Scope and focus	• Scope: Is it defined what part of reality will be described in the model/AD (e.g. only primary processes)? • Aspects: Is it defined what aspects will be described? • The level of detail: Is it defined what level of detail will be described?
	Currency of EA/SA description	• EA description: Degree with which the current version of the documentation is up to date (Percents, subjective evaluation). • EA description: Number of architecture changes made after EA description has been produced. • SA description: Does information reflect a system? • EA and SA descriptions: Frequency with which AD is kept current (number of updates per year).
	Correctness	Verification of information: • Is the information included in an AD/model verified? • Is there any incorrect arguments, or in-accurate or untrue reasoning?
	Sufficiency/ Completeness	• AD's coverage of required viewpoints: The degree to which AD addresses each required architectural viewpoint • Sufficient amount of information: Is the all required information included in the model? Are all topics relating stakeholder's objectives and concerns covered, and only those topics? • Sufficient level of detail: Has each topic has just the detail that stakeholder needs?
	Consistency	• Are the models presenting different viewpoints consistent with each other?
Quality of presentation/ visualization	Conformance to corporate standards	• Does the presentation of the AD/model conform to the corporate standards (if any) for such documents?
	Retrievability: Presentation	• Does the model have an intuitive structure for the stakeholder?

	familiar to stakeholders	
	Retrievability: Notation and structures	• Do models use a defined notation? • Is the notation/structure of model explained? • Is stakeholder familiar with notation?
	Vocabularity and concepts	• Is the vocabulary and concepts stakeholders' concepts? Are the terms and concepts used known by stakeholder? • Are the terms used defined? Are the (new) concepts defined and explained? • Are the names of elements descriptive?
	Complexity: information amount	Is there too much information included in the model? • The number of elements in the model • The number of types of elements in the model • The number of relations depicted in the model • The number of architectural viewpoints
	Complexity: visual complexity	• Proximity: Are the related objects placed near to each other in a model? • Continuity: Is there any right angles positioned next to each other? • Closure: Are objects symmetry and regular? • Similarity: Are similar objects presented in the similar way? • Common fate: Are similar object presented to move or function a similar manner?

Architecture documentation management	Maintenance of ADs and models	• Ownership: Is the staff responsible for AD clearly identified and supported?
		• Maintenance practice:
		  • Is it known how the AD will be maintained once it has been accepted?
		  • Frequency of updates: Number of updates / year or project
		  • Needs for updates: Number of architecture changes made (in a year, in projects) that require documentation update
		  • Maintainability of models: The relative easiness or difficulty with which the documentation can be updated, including revision dates and distribution of new versions and the relative ease or difficulty with which the consistency between descriptions can be checked.

	Cost effectiveness of EA documentation	• Costs: Time and resources needed to produce or update EA descriptions or models: Man-days needed • Amount of documentation: Number of

		documents/models • Frequency of EA documentation updates: Updates / project or updates / year • Needs for updates: Number of architecture changes made (in a year, in projects) that require documentation update

	Cost effectiveness of project architecture documentation	• Costs: Time and resources needed to produce or update project related architecture description or models
		  • Man-days needed
		  • Amount of architectural documentation: Number of documents/models/project
		  • Frequency of updates: Updates / projectNeeds for updates: Number of architecture changes made (in a year, in projects) that require documentation update
		  •

	Architectural framework and views	• Architectural framework: Does an architectural framework for EA/SA exist? Is the framework accepted in the organisation? Is the framework used in the EA/SA documentation work?
		• Architectural views: Are the suitable architectural views chosen for the company or for the project? Relating to each viewpoint are the following aspects defined:
		  • Viewpoint's name?
		  • The stakeholders the viewpoint is aimed at?
		  • The concerns the viewpoint addresses?
		  • The language, modelling techniques, or analytical methods to be used in constructing a view based upon the viewpoint?

	Tools for AD and models	• Support for organisation's framework and viewpoints: Do the design tools support the framework and viewpoints that organisation has chosen to use? Do the design tools support production of the deliverables required? • Suitability for Stakeholders: Is there ability to represent architecture models and views in a way meaningful to stakeholders (e.g. to non-technical stakeholders)? • Repository for architectural documentation: Is there an EA repository for storage and dissemination of the captured EA information?

Top of page

Communication

Communication and a common language can be regarded as one of the main factors helping to succeed in the architectural work (Lankhorst 2005). In order to verify the success in this area contributing to the success of EA in organizations, evaluation of communication and common languag is needed.

Communication studies have been conducted for decades. Even communication audit studies - evaluation of organizational communication (both internal and external) - go back to 1970's and beyond. Communication audits can be carried out in many ways (see e.g. Hargie and Tourish 2000), but the most usual and perhaps the most inexpensive way to evaluate communication is to collect information through a questionnaire. For instance, Downs and Hazen's 'Communication Satisfaction Questionnaire' (Downs, 1988) includes 46 questions. The premise of their work is that the quality and amount of communication in our jobs contribute to both our job satisfaction and our productivity. Another example of questionnaires is presented by Hargie and Tourish's (2000). Their 'Communication Audit Questionnaire' includes 13 sections, each of which many questions or statements.

Based on the above mentioned facts, in our study of evaluation criteria and metrics for architectural communication was to a great extent, an application of communication audit studies. We suggest that evaluation of architecture related communication and common language could be conducted with the help of

- 6 sub-targets in addition to the communication and common language as an evaluation target in its entirety, and
- 13 evaluation criteria in total.

The sub-targets suggested are the following:

- common language/architectural concepts
- communication strategy/plan
- information received through architectural communication
- information sent through architectural communication
- communication channels
- communication skills.

Respectively, the suggested evaluation criteria include, for example,

- Accuracy
- Availability
- Communication Activeness
- Comprehensibility
- Expertise
- Satisfaction
- Timeliness.

Metric examples related to the sub-targets as well as to the communication and common language in its entirety are presented in the table below.

Table: Examples of metrics for evaluating communication and common language.

Evaluation Target	Metric Examples

Common language/ architectural concepts	• Availability: Are the architectural concepts defined and documented? Specifically, has the concept of EA been defined (what does EA mean in the organization)? • Comprehensibility: Are the concepts and terms simple enough, clear and understandable?
Communication strategy/ plan	• Accuracy: Is the communication strategy/plan up-to-date? • Availability: Is the communications strategy/plan available to the key stakeholders (e.g. in a file system or in intranet)? If not, why not?
Information received through architectural communication	• Satisfaction: How satisfied you are with the amount and quality of business information essential for the EA development received from the management/business? (= downward communication) • Timeliness: Extent to which you receive on time the architecture related information needed to do your job.
Information sent through architectural communication	• Satisfaction: How satisfied are you with the amount and quality of information you send to management/business? (= upward communication)
Communication channels	• Availability: Which channels do you use in architectural communication? • Availability: Are these channels easily available? Is the information easily available through these channels?
Communication skills	• Expertise: How satisfied are you with the communication skills of yourself/your co-workers/the architecture team/the management? • Comprehensibility: How understandable and clear is the communication/information provided by the architecture team?
Communication and common language in its entirety	• Communication activeness: How actively are you participating to architecture related discussions/architecture development/architecture related briefings/etc.? • Communication activeness: How actively do you provide architecture related feedback to the architecture team/the management/your co-workers? • Satisfaction: How would you change architectural communication to make you more satisfied? • Satisfaction: How satisfied are you with the communication between the departments/business areas/subsidiaries etc.? (= horizontal communication)

Already the examples in table above show that there is an extensive selection of evaluation questions and metrics for communication and common language. The selection is primarily meant to stimulate and help the definition of the organization specific questions and metrics.

It also seems rational that evaluation of communication and common language are related to the phase of the EA development in the organization or, more specifically, to the EA maturity level of the organization. In different phases or maturity levels, different metrics are used. Most typically, simple metrics are needed in the initializing phase, and

more advanced metrics (e.g. quantitative metrics) can be adopted in later phases.

More detailed information on the evaluation of communication and common language can be found in report Assessing Architectural Work - Criteria and Metrics for Evaluating Communication & Common Language and Commitment.

Top of page

Commitment

The importance of gaining commitment to the EA approach and development can be put as follows:

> "Without a shared sense of purpose and mission, effective governance structure, and executive leadership and commitment, enterprise architecture will only have a minimal impact". (Nelson 2004)

Commitment is also regarded as one of the potential CSFs for EA. In order to verify the success in this area, which also contributes to the success of EA in organizations, evaluation is needed.

We suggest that evaluation of commitment could be conducted with the help of five evaluation criteria:

- Awareness
- Acceptability
- (Customer) satisfaction
- Involvement and participation activeness
- (Adequacy of) resources.

Furthermore, a selection of evaluation questions that demonstrate each evaluation criteria can be used to stimulate the definition of the organization specific evaluation questions/metrics. Examples of the evaluation questions are presented in the table below.

Table: Examples of metrics for evaluating commitment on the EA work.

Evaluation Criteria	Metric Examples
Awareness	• Have you heard/have you been informed about the EA/architecture approach adopted in the organization? • If you have heard about the EA/architecture approach, how satisfied you are with the amount and quality of information you have received?
Acceptability	• To what extent do you consider the EA/architecture approach to be important/useful/essential to the success of the entire organization/your department/your team/your personal work tasks?
(Customer) Satisfaction	• To what extent you utilize architecture guidelines, architecture documentation, or architecture guidance given by architects as a normal part of you work tasks? • What kind of improvement is needed to make you utilize

	the architecture guidelines, documentation, or architecture guidance given by architects more often?
Involvement and Participation Activeness	• Does the EA governance team include executive-level representatives from each line of business? Do they have the authority to commit resources and enforce decisions within their respective organizational units? • How satisfied are you with the extent you participate in architecture development/architectural work development (process development)/architecture management and guidance/architecture implementation projects/architecture related discussions, briefings or training? • What kinds of actions are needed to make you participate in the architecture development, discussions, etc. more often?
Adequacy of Resources	• Does a budget for EA exist? How much funding is directed to the EA development and management/to the entire EA program? • Does a schedule for EA development exist? • Has an architecture team (architects) been assigned? Does a chief architect exist? • Have the architecture team member's responsibilities and authorities been defined? • Has the architecture ownership been defined? • Is the architecture team capable of focusing only to EA/architectural work?

In the beginning of the EA journey, the management's (referring to the top management, CFO, superiors, etc.) commitment to the EA approach is more crucial than the organizational buy-in. This indicates that similar to the evaluation of communication, commitment is also related to the phase of the EA development in the organization or, more specifically, to the EA maturity level of the organization. Thus, the number of committed stakeholder groups should increase as the maturity advances.

It can also be questioned whether commitment needs to be evaluated as a separate target at all. When the EA benefits, and also the success of communication practices, are assessed, it is possible to draw some conclusions about the level of commitment as well. If any benefits cannot be demonstrated, it is likely that no commitment exists either in the organization, or the level of commitment does not increase from the level of awareness. Additionally, if the EA budget exists, it proves at least the commitment of the management.

More detailed information on the evaluation of commitment can be found in the report Assessing Architectural Work - Criteria and Metrics for Evaluating Communication & Common Language and Commitment.

Top of page

EA Compliance

Compliance in general mainly refers to the conformance with rules - standards, regulations, laws, contracts and so forth (Allman 2006; PEER Center 2006; Quality

Assurance Project 2006), but no single established definition seems to exist. The same applies in the EA context as well. As suggested by literature (Aziz et al. 2006; CIO Council 2001; GAO 2003; Spurway & Patterson 2005; The Open Group 2006) and the results of a focus group interview of practitioners, EA compliance can be divided into the following two parts:

- **Internal compliance** refers to the compliance between investments - as well as the projects that implement the investments - and EA with its policies and guidelines.
- **External compliance** is about the compliance between EA and business - are the EA guidelines and target state descriptions in line with the business vision, mission, objectives, strategies, and action plans. External compliance may also refer to EA's ability to react to the changing environment of the organization, as well as to the compliance of EA with the laws and regulations the organization needs to obey.

Organizations typically evaluate EA compliance to fulfill the following goals:

- **Directing a project or an investment to comply with EA - the proactive approach** (adapted from Spurway & Patterson 2005; see also e.g. NIH 2006; Paras 2005; The Open Group 2006): this includes particularly direction and guidance of projects and investments to ensure that the organization is moving towards the target EA, supporting projects and investments by defining how and when EA artifacts are utilized, and encouraging the organization, especially IT projects, to utilize EA descriptions and guidelines.
- **Assuring the compliance between the output of a project or an investment and EA - the reactive approach** (adapted from Spurway & Patterson 2005; see also e.g. GAO 2003; NIH 2006): this includes EA reviews and assessments within projects and investments, and project and investment follow-up with regard to EA descriptions.
- **Assuring the compliance between EA and internal or external standards, reference models and principles** (adapted from The Open Group 2006): this includes evaluation of EA descriptions to be constructed according to defined standards, reference models and principles, by both the organization and external authorities.
- **Ensuring the usability and appropriateness of EA policies, EA frameworks, EA descriptions, business objectives and so forth**: this is highlighted particularly in the cases where compliance requirements cannot be met, possibly suggesting a need to modify EA descriptions, standards, policies and principles, or even business requirements. Also, experience-based feedback can be received from projects and investment processes to improve EA.

Regarding the actual targets of EA compliance evaluations, the high-level objects as well as the evaluation targets of both internal and external compliance, are displayed in the figure below. Compliance between the objects - the evaluation targets - is depicted with arrows. Block arrows depict primary internal or external compliance evaluation targets and small dotted arrows other possible targets to be evaluated. Additionally, examples of lower-level items belonging to each object are included to illustrate the possible documents that can be utilized in compliance evaluation.

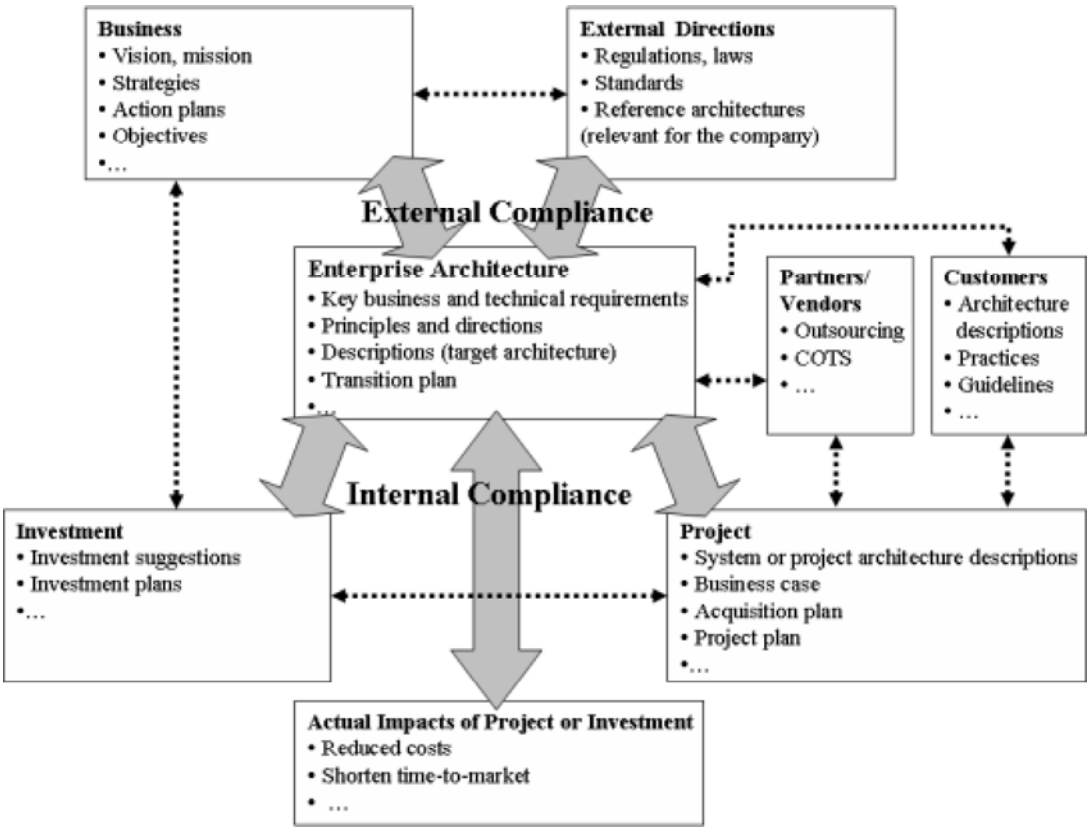


Figure: EA compliance objects and evaluation targets (derived from the focus group interview results).

EA compliance evaluation can be regarded as a part of EA governance. In practice, the persons or teams that have responsibility on the areas of the evaluation objects are suggested to be the evaluators of EA compliance. The EA team is in a key role in EA compliance evaluation by providing guidance and direction to projects and possibly by conducting formal compliance reviews. Business architects and developers, on the other hand, could perform or assist in evaluating the compliance between business and EA. If an EA governance board or EA steering committee exists in an organization (including representatives from various stakeholder groups), it may also have the responsibility of evaluating EA compliance. Thus, possible problems encountered if any single stakeholder evaluates its own work can be avoided.

Typically, the status of EA compliance is illustrated by compliance levels. For example, TOGAF (The Open Group 2006) defines six levels of compliance between architecture specification and its implementation (internal compliance). Departmenf of Defence (BTA 2006), on the other hand, defines three levels of internal compliance. In addition to these metrics, several tools and procedures have been developed to support carrying out EA compliance evaluation (Eurocontrol 2006; NIMA 1998; The Open Group 2006). However, each organization needs to make its own decisions on the actual steps of the evaluation process, and to implement it as a continuous EA governance activity.

Finally, when planning and conducting EA compliance evaluation, it should be noted that because organizations' environment is constantly changing and so are their EAs, EA compliance has a dynamic nature. Therefore, compliance can be evaluated to be on an acceptable level at the moment, but it does not guarantee that this is the case in the future. EA compliance also seems to depend on the EA maturity level. In the lower levels of maturity (in the beginning of EA development work), EA compliance and its evaluation actually equals quality assurance, and especially the impacts of EA work are a

focal issue. After the EA process has become more established, more profound aspects of EA compliance will become increasingly important. For more information on EA compliance and its evaluation, see report Evaluating Enterprise Architecture Compliance and paper Enterprise Architecture Compliance: the Viewpoint of Evaluation.

Top of page

Business-IT Alignment

Alignment between business and IT has been considered important in organizations for over 15 years (Luftman 2000). As a high degree of alignment has been associated with improved business performance by empirical evidence (Chan et al. 1997; Papp 1999), it is not surprising that business-IT alignment has been continuously considered as one of the top concerns of company executives such as CIOs (Luftman et al. 2006) and a great number of studies have been conducted on the subject so far (Chan 2002; Luftman 2000). Alignment has also been considered as one of the key benefits or potential objectives of EA (Goethals et al. 2006; Kluge et al. 2006; Ross & Weill 2005).

According to literature, alignment between business and IT is an evolutionary process (Avison et al. 2004; Cumps et al. 2006; Maes et al. 2000), which needs to be maintained over time by planning, design, management, and evaluation activities on both strategic and tactical levels (Hu & Huang 2005; Maes et al. 2000). Moreover, alignment may refer to the extent or amount of alignment, measured by e.g. various maturity models (see e.g. Luftman 2000; Reich & Benbasat 2000). Despite the large number of models developed to depict this complex phenomenon (see e.g. Chan et al. 1997; Cumps et al. 2006; Luftman 2000, the Strategic Alignment Model (SAM) (Henderson & Venkatraman 1993) remains the most commonly referred.

In general, several common factors affecting alignment can be derived:

- **strategic factors**, such as business and IT strategies, plans, objectives and vision (e.g. Avison et al. 2004; Henderson & Venkatraman 1993; Luftman 2000; Reich & Benbasat 1996)
- **structural factors**, such as processes, organizational structure, architectures, governance and competences (e.g. Chan 2002; Maes et al. 2000; Weiss & Anderson 2004)
- **social and cognitive factors**, such as communication, partnership, learning, and common knowledge and understanding (e.g. Ciborra 1997; Luftman 2000; Weiss & Anderson 2004)
- **measurement and evaluation factors**, such as metrics and measurement systems for both business and IT (e.g. Luftman 2000; Moody 2003).

Alignment, in turn, is argued to lead to a multitude of benefits, of which several have been empirically substantiated (see e.g. Chan 2002; Papp 1999). Practically, alignment is suggested to be the responsibility of IT governance (Dahlberg & Kivijärvi 2006; Symons 2005), which in turn needs to be in close relationship with business. However, since research offers little contributions to practice (e.g. Cumps et al. 2006; Maes et al. 2000), alignment remains challenging to improve, sustain, or evaluate in practice. Moreover, factors affecting alignment encompass the entire organization, indicating that an extensive, holistic approach would be needed to address these issues.

It has been suggested that EA could be this kind of an approach (c.f. Hirvonen & Pulkkinen 2003; Morganwalp & Sage 2004), but the relationship between EA and business-IT alignment is more complex than this viewpoint alone. As brought out,

- EA can be regarded as an enabler of improved alignment in organizations (e.g. Goethals et al. 2006; Kluge et al. 2006; Ross & Weill 2005), by providing tools for describing and communicating various aspects of an organization (e.g. the business strategy and objectives), as well as for achieving ISs that support the business.
- The factors affecting the success of EA and the extent of alignment are somewhat similar, even implying that alignment improvement efforts can be regarded as EA work having a slightly different scope and emphasis.
- Alignment in the EA context refers to the concept of EA compliance. The EA compliance evaluation objects may also be the potential objects between which alignment is needed in the EA context to enable the organization to reach alignment between business and IT. However, compliance does not guarantee alignment.
- Alignment in the EA context may refer to the alignment between various architectures, or architectural views, of an organization (c.f. Chen et al. 2005; Pereira & Sousa 2003).
- EA maturity (see the section EA evaluation model) and business-IT alignment correlate, but do not explain one another (van der Raadt et al. 2005).

Various approaches exist for evaluating business-IT alignment. What seems to differentiate these approaches from each other is that they have a slightly different focus on the issues to be evaluated. They also seem to provide metrics of different granularity compared to each other. Both Luftman (2000; 2003) and Reich & Benbasat (2000) provide a wide selection of evaluation metrics, ranging from soft issues (e.g. communication) to hard issues (e.g. business metrics or skills-related metrics). The soft aspects, especially the communication point of view, seems to be missing from the examples provided by Chan et al. (1997) and Symons (2005). Combination of both qualitative (soft) metrics and quantitative (hard) metrics should be implemented to develop a comprehensive measurement instrument for business-IT alignment. In the EA context, the same evaluators that evaluate EA compliance could potentially be the evaluators of alignment as well. An organization may also need a functional EA governance board which is responsible for evaluating alignment periodically (Jayashetty et al. 2004).

For more information on business-IT alignment and its evaluation in general and specifically in the EA context, see report Evaluating Business-IT Alignment in the Enterprise Architecture Context.

Top of page

Benefits of Architecture Work

Even though literature suggests that a multitude of benefits can be realized by architectural work, there is little empirical evidence. This may be due to the extensiveness of architecture, architecture work and their impacts, and their state of constant change that make it difficult to evaluate the benefits and, more importantly, attribute them to architecture work (see Hjort-Madsen 2006; Kamogawa & Okada 2005). However, the need for evaluating the benefits is evident because it provides a rationale for the key stakeholder support and investments in architecture work (see Infosys 2005; Ross & Weill 2005; Schekkerman 2005).

Still, literature on architecture work benefits is rare and focuses almost entirely on referring to a number of achievable benefits without solid evidence. Nearly thirty different benefits can be derived from literature, but they are typically not clearly defined

in literature, their levels of abstraction differ, and their interrelationships remain largely unknown. Moreover, it is not known from which parts or characteristics of architecture or architecture work benefits essentially realize. This implicates that there is a need for describing the architecture work benefit realization process which in turn facilitates the creation of a comprehensive architecture work evaluation model. Currently, no freely available validated models exist, but a few case studies (see e.g. Luftman 2000) and survey-based studies (Rosser 2006; Saha 2006; Schmidt 2005) have been made to present realized benefits of architecture work. In related contexts, such as management and IT, a large number of metrics have also been suggested and could be used to measure individual benefits such as business-IT alignment. Also, guidelines for measuring Return on Investment (ROI) in the architecture context have been presented.

To disentangle the myriad of proposed benefits and metrics a study including a literature review and a focus group interview of practitioners was carried out in the AISA project. Because no guiding evaluation model existed, metrics and evaluation criteria were charted from literature and assigned directly to the architecture work benefits derived from literature. Each of the benefits was complemented with up to 60 metrics. However, the focus group brought out that that the metrics presented were too great in number and would not suit practice without a guiding evaluation model. In turn, the focus group approached the problem from a practical perspective and proposed three main categories into which the proposed architectural work benefits could first be categorized:

- costs,
- growth and
- flexibility.

The categories are based on the basic targets and needs of a business enterprise and its owners. Then, the group suggested a practical view of the architectural work benefits and their evaluation using the three categories of architectural work benefits as a basis for constructing architectural work and corporate evaluation and measurement system. The view takes into account three viewpoints of evaluation:

- corporate metrics consulted by the architecture team,
- metrics of the architectural work itself, and
- metrics of architectural work results.

It illustrates

- corporate level targets (the three architectural work benefit categories),
- layered hierarchy of metrics,
- relationships between architectural and corporate metrics,
- architecture team/unit role and position, and
- role of architectural work ROI.

The practical view is depicted in the figure below.

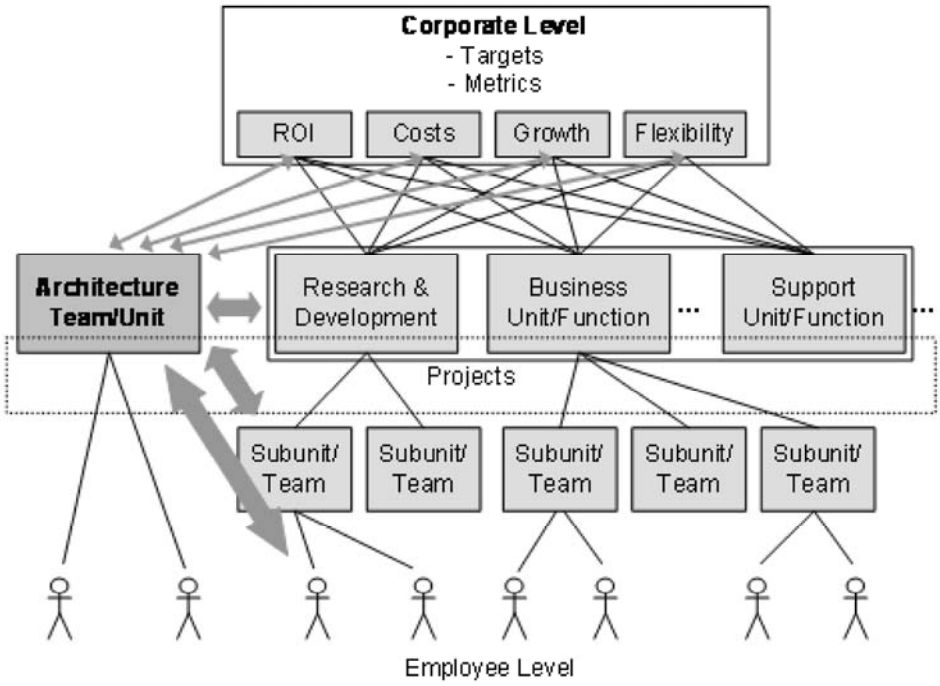


Figure: The practical view of architectural work benefits and their evaluation (developed by the focus group).

The practical view depicts a hierarchy of metrics used in an organization, starting from the corporate level, where metrics for the enterprise's most important targets, such as costs, growth and flexibility, are implemented. From there, management implements the metrics derived from the top level targets to the unit or function level below. From the unit or function level, middle management implements metrics for subunits or teams of employees, and from there, metrics are implemented to individual employees. In addition, projects usually have their own metrics as well as the architecture team or unit. For each unit, function, subunit, team and individual, 3-5 metrics should be implemented. In addition to implementing the metrics from top to bottom, feedback from bottom to top is also needed to preserve the links and compatibility between the metrics on adjacent levels. From the metrics in the hierarchy, management should be provided with 3-5 metrics which can be used to evaluate architectural work benefits. Because of the hierarchy, the architecture team or unit can rationalize that benefits are received from architectural work in the organization's functions and units. A ROI metric for architectural work should also be implemented to measure whether the architectural work carried out is profitable in the long-term.

Managing the integrity of the measurement system as a whole is vital. The hierarchy of metrics should be low enough to preserve the chain of causalities between the metrics on adjacent levels. If the hierarchy grows too high, it may result in inconsistent metrics on the lower levels of the hierarchy. The size of the hierarchy is dependent on the size of the enterprise, 5-6 levels would be a feasible example. The implemented metrics should also be selected carefully, taking into account the goals of the evaluation, especially the guiding effect of evaluation on individuals, teams, units and functions. In addition, it is essential that architectural work metrics are connected to other organizational metrics.

The focus group identified a few additional challenges related to architectural work benefit evaluation. Firstly, a baseline or standard for evaluation results does not typically exist in organizations since especially EA is a new discipline. Secondly, it is challenging to find a mutual understanding of the time scale of presenting benefits between

management and the architecture team, and a balance between producing short-term and long-term benefits. On one hand, quick wins are essential in gaining management support, but on the other hand, architecture work is long-term in nature. Finally, the focus group agreed that the results of architecture work evaluation should also be communicated in the organization, taking into account 1) what is to be communicated, 2) to whom the communication is aimed, and 3) when is the right time to communicate (see the section on communication for more information).

For more information on architecture work benefit evaluation, see report Evaluating the Benefits of Architectural Work. For information on the benefits of architecture work in general, see section on the benefits of the architectural work and paper Enterprise Architecture Benefits: Perceptions from Literature and Practice.

Top of page Page Index

Authors: Eetu Niemi, Niina Hämäläinen & Tanja Ylimäki

Decision-Making and Risk Management in Architecture Work

On this page, the following two topics are addressed:

- Architectural Decision-Making: Introduction to decision-making in architecture work
- Architectural Risks and Architectural Risk Management: Introduction to risks encountered in architecture work

Architectural Decision-Making

Decision making is generally understood as a cognitive process leading to the selection of a course of action among alternatives. It begins when a need to do something exists but the course of action is not yet clear. The **decision making process** involves

- definition of the problem,
- information gathering,
- identification and evaluation of alternatives,
- the actual decision,
- its implementation, and
- follow-up assessment (Power 2002).

In reality the process is not an explicit cycle but iteration is done between and during tasks. A decision may set a course for follow-up decisions, initiating a new decision making process. Decisions are also made on various levels in organizations including

- strategic management,
- IT portfolio management and
- project management.

Finding the right information to make the decision is a constant challenge and typically the party that collects the information and presents the alternatives is different to the actual decision maker. Solution alternatives are evaluated with a certain decision making criteria. In **rational decision making**, alternatives and their consequences have to be known, decision makers must have a set of preferences to guide decision making, and a decision rule has to be utilized to select a single alternative on the basis of its consequences for the preferences (Shapira 1997).

In the architecture context, decisions are typically related to architectural plans and models. A complex architecture can reflect thousands of decisions (Tyree & Akerman 2005). Architecture decisions relate to different architectural levels (e.g. enterprise, domain, application and component architecture) and should only define elements on that specific level (Malan 2002).

On **the level of EA**, architecture decisions are high level decisions that can involve

- Selection of architecture plans (target, transition, vision)
- Selection of architecture standards, principles and guidelines

- Decisions about the objectives of architecture work in the organization.

In **the SA domain**, architecture decisions are typically made on the level of architectural models and typically define (see Bass et al. 1998)

- a system's key structural elements,
- the externally visible properties of these elements and their relationships, and
- how to achieve the architecturally significant needs and requirements.

When making architectural decisions it should be first considered whether the decision actually is absolutely necessary to achieve business strategy and meet architectural objectives (Malan & Bredemeyer 2002; 2004). Thus, the decision should be traceable to business objectives and not be overly detailed on the enterprise level (Malan & Bredemeyer 2002; 2004). It is futile to make decisions if they are not enforceable and actually enforced (Malan & Bredemeyer 2002; 2004). Decisions should also take into account possible change needs to the architecture in the future by building in agility to the architecture. To enable follow-up evaluation of decisions and increase stakeholder commitment to them, decisions should be communicated to the stakeholders with their rationale. Decisions may be documented to share them with stakeholders but merely storing documentation in a repository is not sufficient.

In the architecture context, decision making points and levels are dependent on the organization in question. Some organizations do not want to establish new decision making points for architecture, so architectural decision making may be fragmented to various existing decision making points. Since architectural decisions are made on many levels in the organization, decision makers involve various roles including architects, project roles and business management roles. Architecture may have influence on other decision in the organization (e.g. IT portfolio and project planning) but on the other hand, some architectural decisions may need approval from various other decision making points (e.g. business or IT management).

It should be noted that architecture is only one way of enforcing organizational strategies and thus business may have a first say when planning organizational improvements. Business may even make architecture decisions without considering the big picture. Therefore, architects need to communicate with stakeholders to find a common ground.

All in all, architectural decision-making is similar to other kinds of organizational decision-making. Therefore it should be carefully considered whether or not separate decision-making mechanisms are needed for architecture decisions.

For more information on the subject, see report Architecture Planning and Decision Making in Companies.

Top of page

Architectural Risks and Architectural Risk Management

Because EA is an extensive program, it requires considerable investments and may thus result in many political, project management and organizational challenges (Kaisler et al. 2005). As with any investment, also EA investments (investments related or driven by EA) involve risks which need to be identified and managed (Saha 2006). Organizations investing in EA may face unexpected materialized risks related to business and ICT alike, threatening the success of the EA program. Moreover, since EA is a critical management

tool materialized risks can have serious consequences in the organization utilizing EA.

The extensive, continuous and iterative nature of the EA approach further complicates EA risk identification and management. Unpredictable effects may arise from EA processes (e.g. planning, development, management, maintenance and use) or may be associated with any of the levels of architectural models of the organization (e.g. business, information, information systems, technology) (Baldwin et al. 2007). However, literature on risks in the context of EA is rare even though risks have been extensively discussed in generic risk literature (see e.g. Crouhy et al. 2001; Lam 2003; Reuvid 2005) and related contexts such as IT and IS (see e.g. Boehm 1991; Benaroch 2002; Sherer and Alter 2004; Keyes 2005; Benaroch et al. 2006). In the EA context, work on EA investment risks and options has been conducted (see Saha 2006).

The Collins English Dictionary defines risk as

> "the possibility of incurring misfortune or loss".

However, in risk literature many authors do not even provide a definition for the term. This may be partly explained by the complex nature of risks. First, they have many characteristics such as

- exposure (maximum amount of damage suffered),
- severity (amount of damage that is likely suffered),
- volatility (variability of potential outcomes),
- probability (how likely a risky event occurs),
- time horizon (the time exposed to the risk),
- correlation (amount of correlation between different risks) and
- capital (how much capital is needed to cover losses) (Lam 2003).

Second, all risks are temporal and can thus be materialized in complex chains of risks and mitigations over time (Alter and Sherer 2004). Third, risks are not always negative but may also have positive consequences when they materialize (Alter & Sherer 2004). Risks can also be conceptualized in a number of different ways. We suggest that **EA risks** are

- any factors that may lead to negative outcomes in the EA program, and
- any negative outcomes resulting from these factors.

A large number of risks have been uncovered in literature and different classifications for them have been proposed. **The work system framework of risks** (see Sherer and Alter 2004) is adapted for the classification of EA risks because of its genericity and extensive literature base. Generic work system risks apply to the IS context (Sherer and Alter 2004), suggesting that they may apply to the EA context as well. The framework includes nine elements which all contribute to the operation of the system. For each of the elements, a number of risks are adapted from literature (see Sherer and Alter 2004), complemented by the practical experiences of the participants of a focus group interview of practitioners.

Regarding to EA risk management, it should be noted that even though EA-related risks are not currently considered in detail in organizations, there seems to be the need of managing them. EA risk management is not an independent 'island' in an organization; on the contrary, it should be in a close connection or a part of organizational risk management. In turn, EA facilitates organizational risk management. In EA-related decision-making, risk management activities should be utilized to optimize the risk-gain

ratio, and decision follow-up implemented as a continuous activity. To control EA risks, communication, common language and sufficient EA documentation are important activities. In addition, EA risk management responsibilities should be clearly defined, also more extensively than on the level of one architectural development project.

For more information on the subject, see report Enterprise Architecture Risks - An Overview.

Top of page Page Index

Authors: Tanja Ylimäki and Eetu Niemi

Case Studies

In this section, we shortly describe the case studies that were conducted in three companies participating in the AISA project:

- Case Elisa
- Case IBM
- Case Osuuspankkikeskus (OPK)

The cases were twofold:

- First, the companies were considered as research targets: a study was conducted to chart the status (maturity) of architecture work in the first and the last year of the project. The results of these case studies are described in section Status of Architecture Work
- Second, some of the research results, especially the metrics and criteria for evaluating architecture work, were applied in the case companies.

Case Elisa

Primary Participants at Elisa	• Ari Andersin • Juha Jantunen • Merja Kalttonen • Pekka Karppinen • Jarkko Lahtinen
Primary Objectives	• To chart the status (maturity) of the architecture work at Elisa • To analyze how the architecture work has evolved and matured during the research project • To provide support for developing the architecture work practices at Elisa, especially in the area of evaluating the architecture work and its outcomes
Main Results	The confidential company-specific reports related to the case Elisa dealt with • the current status of architecture work at Elisa, year 2006 • the current status of architecture work at Elisa, year 2008 • the measurement and evaluation of architecture work and architectures at Elisa See also the paper 'Enterprise Architecture Process of a Telecommunication Company - A Case Study on Initialization' in the Results section.
Lessons Learned	• Architecture work has been initialized successfully and it is progressing. • Measurement and evaluation of the progress and the gained benefits of architectures and architecture work is also an important issue that cannot be neglected.

	• The adaptation of architecture framework should be done on the basis of the organization and business culture of the company. 'Ready-to-use' frameworks simply do not exist. • It is not a quick process to build an EA culture in an organization.
Further Development	• Selecting the most relevant metrics and criteria for evaluating architecture work and architectures from the suggested set of metrics. • Putting the evaluation and measurement activities into practice in line with the existing performance measurement system in the organization. • Continuing the development of enterprise architecture practices and architectures as suggested e.g. by the company-specific architecture work status reports and existing in-house architecture roadmaps. • Improving the co-operation between enterprise architecture, projects and business management. • Developing an efficient and easy method to communicate on EA.

Top of page

Case IBM

Primary Participants at IBM Finland	• Petri Ahveninen • Stina Carlsson • Kimmo Kaskikallio • Markus Kinni • Jouko Poutanen

Primary Objectives	• What is the status of
	  • the architecture work practices IBM Finland can provide to its customers and
	  • the architecture work practices of IBM customers as perceived by consultants and architects?
	• What kind of tools or practices would enable a consultant or architect
	  • to identify and document the status of architecture work in a customer organization or
	  • to support the customer to understand its own status in architecture work?

Main Results	The confidential company-specific reports related to the case IBM dealt with: • the status of architecture work, year 2006 • the checklist for charting the architecture work status of a customer organization
Lessons Learned	• The architecture work status study revealed that IBM has a lot of methods and practices, as well as knowledge and training possibilities for enhancing architecture work of its

	customers, but the customers are not necessarily ready to adopt these practices. Therefore, IBM could aim at increasing the architecture awareness of the customers, e.g. by accentuating the expected benefits of the architecture work in a particular customer case. • The checklist developed in the project can be utilized as one possible tool to increase the customer's architecture awareness by indicating a wide selection of issues important in architecture development and management.
Further Development	• The checklist developed in the project needs to be tested in practice to evaluate its usability. Based on the test results, corrections and additions may be required.

Top of page

Case Osuuspankkikeskus (OPK)

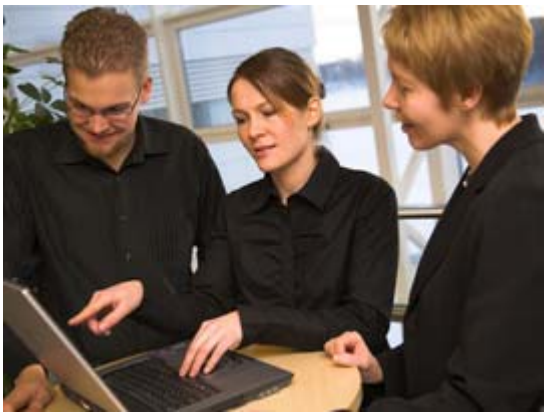
Primary Participants at OPK	• Markku Korhonen • Jouni Lähteenmäki • Jari Vänskä • Kari Makkonen • Heikki Salo
Primary Objectives	• To chart the status (maturity) of the architecture work at OPK • To analyze how the architecture work has evolved and matured during the research project • To provide support for developing the architecture work practices at OPK, especially in the area of evaluating the architecture work and its outcomes
Main Results	The confidential company-specific reports related to the case OPK dealt with: • the current status of architecture work at OPK, year 2006 • the current status of architecture work at OPK, year 2008 • the measurement and evaluation of architecture work and architectures at OPK
Lessons Learned	• As the architecture team is the bridge between the business and ICT, both up-ward communication (with the business) and down-ward communication (with the project architects and project managers) are essential. Especially, the communication between the architecture team and the project architects and project managers has been successful and the awareness, as well as the actual use, of the available architecture guidance has increased. • Measurement and evaluation is conducted at some architectural levels, but evaluation of the whole EA program and its benefits will be more relevant as the architecture work will become more stabilized and mature.
Further	• Continuing the development of architecture work practices

Development	and architectures e.g. according to the suggestions provided by the company-specific reports and the existing in-house plans

Top of page Page Index

Results

This section summarizes the results of the AISA project. For an overall discussion, see the Lessons Learned section. After a brief overview, the reports are listed below, organized by type in separate tables.



Overview of the Results

The project has reached the main objectives that were set: to study the characteristics of both the architecture planning and development processes of high quality and maturity and the EA and SA of high quality and maturity. These research themes were clarified with the help of several sub-topics (constituting the main contents of this publication).

The multifaceted nature of architecture work was clarified with the help of

- success and failure factors for SA to provide an extensive list of issues relevant in successful SA development and management (paper information),
- potential critical success factors for EA to provide an extensive list of aspects requiring attention in successful EA work (paper information),
- charting the various stakeholders of architecture work and their typical needs to support the practitioners in planning the EA work to better respond to the requirements of the stakeholders and to better manage the variety of stakeholders (paper information),
- charting the possible benefits of architecture work to provide support for organizations to define the objectives and expected benefits for their architecture programs (paper information),
- charting the status of architecture work in the case organizations by using the evaluationg model constructed in the project (paper information, report information), and
- studying the quality and quality management aspects of architecture work on both the SA (paper information, report information) and EA (report information) levels.

Research related to **the evaluation of architecture work** dealt with charting

- the possible needs and triggers for evaluation in organizations (paper information),
- the various evaluation aspects in architecture work (paper information), and
- the status of evaluation practices in the architecture work (dissertation information, paper information).

On a more practical level, **evaluation practices** were studied and especially

- the essential components of evaluation planning were defined to support the evaluation planning activities in organizations (paper information, report information),
- a generic evaluation model for EA was constructed to provide a tool to analyze the status of architecture work in organizations (paper information), and
- the existing architecture evaluation methods were charted to provide organizations information on the available tools for different kind of evaluation needs in organizations paper information, report information).

Because the architecture work is such a large area and **many evaluation targets** exist, the project had to focus on few evaluation targets for further scrutiny. These targets were the following:

- Architecture documentation: Metrics and criteria were charted for evaluating the quality of architecture documentation (paper information).
- Architecture related communication: Metrics and criteria were charted for evaluating the quality of architecture related communication (report information).
- Commitment to the architecture work: Metrics and criteria were suggested for evaluating the commitment to the architecture work (report information).
- Architecture compliance: A study was conducted to clarify the concept of EA compliance and its evaluation (paper information).
- Business-IT alignment: A study was conducted to clarify the concept of business-IT alignment in the EA domain and its evaluation (report information, paper information).
- Benefits of architecture work: A study was conducted to chart metrics for measuring the many potential benefits of architecture work. As the number of potential metrics is overwhelming, an evaluation model for the benefits was proposed (report information).

Finally, risks (report information) and the decision-making (report information) in architecture work were briefly addressed to provide a theoretical view to these issues to support the planning of risk management and decision-making practices in organizations.

It should be noticed that many of the above topics are creditably covered by the dissertation Evaluation and Measurement in Enterprise and Software Architecture Management.

In the Lessons Learned section, the significance of the results and the overall contribution of the project is discussed.

Scientific Publications

The published papers are presented in aplhabetical order. Due to copyright issues some of the papers cannot be included in this CD-rom publication.

Title	Author(s)	Where and When Published
A Framework to Support Business-IT Alignment in Enterprise Architecture Decision Making.	Hämäläinen Niina and Liimatainen Katja	In Proceedings of the EBRF 2007 Conference 'Research Forum to Understand Business in Knowledge Society', September 25-27,

		2007, Jyväskylä, Finland.
A Goal-Oriented Way to Define Metrics for Enterprise Architecture Program.	Hämäläinen Niina and Kärkkäinen Tommi	In Journal of Enterprise Architecture (vol.4, nr. 1), 2008.
Analysis of the Current State of Enterprise Architecture Evaluation Methods and Practices.	Martin Hoffmann	In Proceedings of the European Conference on Information Management and Evaluation (ECIME 2007), September 20-21, 2007, Montpellier, France.
Architectural Work Status: Challenges and Developmental Potential - A Case Study of Three Finnish Business Enterprises.	Niemi Eetu	In Proceedings of the 6th WSEAS International Conference on Applied Computer Science (ACS'06), December 16-18, 2006, Puerto de la Cruz, Tenerife, Spain.
Defining Enterprise Architecture Risks in Business Environment.	Niemi Eetu and Ylimäki Tanja	In Proceedings of the EBRF 2007 conference 'Research Forum to Understand Business in Knowledge Society', September 25-27, 2007, Jyväskylä, Finland.
Enterprise Architecture Benefits: Perceptions from Literature and Practice.	Niemi Eetu	In Proceedings of the 7th IBIMA Conference on Internet & Information Systems in the Digital Age, December 14-16, 2006, Brescia, Italy.
Enterprise Architecture Compliance: The Viewpoint of Evaluation.	Ylimäki Tanja, Niemi Eetu and Hämäläinen Niina	In Proceedings of the European Conference on Information Management and Evaluation (ECIME 2007), September 20-21, 2007, Montpellier, France.
Enterprise Architecture Evaluation Components.	Niemi Eetu and Ylimäki Tanja	In Proceedings of the 11th International HAAMAHA Conference, July 9-12, 2007 Poznan, Poland.
Enterprise Architecture Process of a Telecommunication Company - A Case Study on Initialization.	Andersin Ari and Hämäläinen Niina	In Proceedings of the 11th International HAAMAHA Conference, July 9-12, 2007 Poznan, Poland.
Enterprise Architecture Stakeholders - A Holistic	Niemi Eetu	In Proceedings of the 13th Americas Conference on

View.		Information, August 9-12, 2007, Keystone, Colorado, USA.
Enterprise Architecture Work Overview in Three Finnish Business Enterprises.	Niemi Eetu	In WSEAS Transactions on Business and Economics (vol. 3, nr. 9), 2006.
Potential Critical Success Factors for Enterprise Architecture.	Ylimäki Tanja	In Journal of Enterprise Architecture (vol.2, nr. 4), 2006.
Quality Evaluation Question Framework for Assessing the Quality of Architecture Documentation.	Hämäläinen Niina and Markkula Jouni	In Proceedings of the International BCS Conference on Software Quality Management (SQM 2007). August 1-2, 2007, Tampere, Finland.
Quality Management Activities for Software Architecture Process.	Hämäläinen Niina	In Proceedings of the IASTED International Conference on Software Engineering (SE 2007), February 13-15, 2007, Innsbruck, Austria.
Success and Failure Factors for Software Architecture.	Hämäläinen Niina, Markkula Jouni, Ylimäki Tanja and Sakkinen Markku	In Proceedings of the 6th IBIMA Conference on Managing Information in the Digital Economy, June 19-21, 2006, Bonn, Germany
The Role of Architecture Evaluations in ICT-companies.	Hämäläinen Niina, Ylimäki Tanja and Niemi Eetu	In Proceedings of the International Business Information Management Conference (6th IBIMA), June 19-21, 2006, Bonn, Germany.
Towards a Generic Evaluation Model for Enterprise Architecture.	Ylimäki Tanja	In Journal of Enterprise Architecture (vol. 3, nr. 3), 2007.

Reports

AISA project reports are presented in alphabetical order.

Title	Author(s)	Type and date
Architectural Work Status: Challenges and Developmental Potential. A Case Study of Three Finnish	Niemi Eetu	report, 30.8.2006, 21 p.

Business Enterprises.		
Architecture Evaluation Methods.	Hoffmann Martin	report, 2.5.2007, 64 p.
Architecture Planning and Decision Making in Companies.	Niemi Eetu and Hämäläinen Niina	report (slides), 6.3.2008, 39 p.
Assessing Architectural Work - Criteria and Metrics for Evaluating Communication & Common Language and Comment.	Ylimäki Tanja	report, 9.2.2007, 37 p.
Bibliography	Niemi Eetu, Ylimäki Tanja,Hoffmann Martin and Hämäläinen Niina	report, 13.8.2007, 93 p.
Enterprise Architecture Risks - An Overview.	Niemi Eetu & Ylimäki Tanja	report, 6.3.2008, 23 p.
Evaluating the Benefits of Architectural Work.	Niemi Eetu and Ylimäki Tanja	report, 26.3.2007, 33 p.
Evaluating Business-IT Alignment in the Architecture Context.	Niemi Eetu and Ylimäki Tanja	report, 4.12.2007, 27 p.
Evaluating Enterprise Architecture Compliance.	Ylimäki Tanja, Niemi Eetu and Hämäläinen Niina	report, 19.4.2007, 20 p.
Evaluation Needs for Enterprise Architecture.	Ylimäki Tanja and Niemi Eetu	report, 18.10.2007, 38 p.
Measurement in Enterprise Architecture Work - The Enterprise Architecture Team Viewpoint.	Hämäläinen Niina, Niemi Eetu and Ylimäki Tanja	report, 16.3.2007, 16 p.
Quality Evaluation of Architectural Documentation and Models.	Hämäläinen Niina	report, 19.12.2007, 30 p.
Quality Management Activities for Enterprise Architecture.	Ylimäki Tanja	report, 4.5.2006, 27 p.
Quality Management Activities in Software Architecture Process.	Hämäläinen Niina	report, 3.5.2006, 18 p.
Success and Failure Factors for Software Architecture.	Hämäläinen Niina, Markkula Jouni, Ylimäki Tanja, Sakkinen Markku	report, 11.1.2006, 25 p.

The Role of Architecture Evaluations in ICT-companies.	Hämäläinen Niina, Ylimäki Tanja, Niemi Eetu	report, 1.11.2006, 23 p.
Towards Critical Success Factors for Enterprise Architecture.	Ylimäki Tanja	report, 11.1.2006, 35 p.

Presentations

Presentations related to AISA Project and its results are provided in alphabetical order.

Title	Author(s)	Date
AISA Research Project. Quality Management of Enterprise and Software Architectures.	Niemi Eetu & Ylimäki Tanja	20.2.2008
Architecture Evaluation Methods.	Hoffmann Martin	18.4.2007
Architecture Planning and Decision Making in Companies.	Niemi Eetu and Hämäläinen Niina	6.3.2008
Architectural Work Status - A Case study of Three Finnish Business Enterprises.	Niemi Eetu	-
Assessing Architectural Work - Criteria and Metrics for Evaluating Communication, Common Language & Commitment.	Ylimäki Tanja	7.2.2007
Enterprise Architecture Compliance Evaluation.	Ylimäki Tanja, Niemi Eetu and Hämäläinen Niina	18.4.2007
Enterprise Architecture Risks - an Overview.	Niemi Eetu and Ylimäki Tanja	20.2.2008
Evaluation Needs for Enterprise Architecture.	Ylimäki Tanja	1.11.2006
Evaluating the Benefits of Architectural Work.	Niemi Eetu	7.2.2007

Evaluating Business-IT Alignment in the EA Context.	Niemi Eetu and Ylimäki Tanja	18.4.2007
Long-Term and Short -Term Architecture Decisions.	Hämäläinen Niina	-
Measurement in Enterprise Architecture Work.	Hämäläinen Niina, Niemi Eetu and Ylimäki Tanja	-
Quality Management Activities for Enterprise Architecture.	Ylimäki Tanja	3.5.2006
Quality Management Activities in Software Architecture Process.	Hämäläinen Niina	-
Role of Architecture Evaluations in ICT-Companies.	Hämäläinen Niina	-
Towards Critical Success Factors for Enterprise Architecture.	Ylimäki Tanja	11.1.2006

Theses

Title	Author(s)	Type and date
Evaluation and Measurement in Enterprise and Software Architecture Management. Keywords: enterprise architecture, software architecture, evaluation, measurement, metric	Hämäläinen Niina	Dissertation, Department of Mathematical Information Technology, University of Jyväskylä, 2008. To be publicly defended on April 3, 2008.
Quality Evaluation of Software Architecture with Application to OpenH.323 Protocol Keywords: quality control, quality attributes, metrics, software architecture evaluation, ATAM, telecommunication, H.323	Hoffmann Martin	Master's Thesis, Department of Mathematical Information Technology, University of Jyväskylä, 2007.

Provided by the Hasso Plattner Institute (University of Potsdam, Germany) in pdf format.		

Top of page Page Index

Lessons Learned

In this section, we summarize the AISA project by addressing the following topics:

- General Conclusions: Which conclusions can be drawn from the three-year research project?
- Implications for Practitioners: How can practitioners utilize the project results?
- Future Trends: How will the architecture work evolve in the future?
- Further Research: Which future research questions arouse?

General Conclusions

In the AISA project, we studied the quality management of both enterprise and software architectures, especially from the viewpoint of evaluation and measurement. From our study, we draw the following conclusions.

Both **EA and SA success factors** defined in the project (see the section on architecture work) present a wide view of factors that affect and should, therefore, be taken into consideration in architecture work. These factors can also be regarded as possible evaluation targets, for which criteria, metrics and methods can be developed. The potential critical success factors for EA also enabled us to construct an initial generic maturity model for EA and use it to evaluate the architectural work status of organizations. All of the test cases demonstrated that the model is comprehensible and usable and that it provides an extensive view on the state of the organization's EA or its ability to support EA development and management in its customer projects. Improvement needs were also detected (see Further Research).

Quality of EA is a concept that does not yet have an established definition (see the section on quality management). To put it simply, we suggest that an EA has high quality if it is understood, accepted and used, and the EA is measured in order to ensure that the quality requirements are met. Furthermore, we consider the maturity models as one means of advancing the quality of EA. However, the quality management activities related to EA are still unclear and further research is, therefore, required.

Instead in the SA domain, quality management was clarified in this project and a **quality model for both the software architecture management process and the software architecture** was constructed (see the section on quality management in the EA and SA context). Furthermore, it seems that **architecture management** is spread out to many processes in organizations. The activities that aim to drive and control the architecture and architectural quality, may be included in several separate processes in organizations, such as investment planning, project management, process management or system development process. Because architecture management processes are not so clearly separate processes in organizations, the capability assessment of architecture management is rather difficult. Therefore, the different processes in organizations also affect the architectures and architectural quality. There also seems to be a need

- to shift from architectures that are driven by investment planning and system development towards architectures driven by architecture management
- for architecture management practices and process models that aim at high-quality architectures as well as a need to advance the maturity (and the quality) of

architecture management processes themselves
- for agility in architecture management and development. Architecture management and development processes cannot be too heavy (e.g. require a lot of time and resources), because the restricted time and quick changes in the environment of the organizations may require changes in them, too
- for metrics and metric programs for evaluating e.g. the architectural maturity, quality and performance.

Generally, **architectural work is currently under development or in initial state**. The results of the case studies (see the section on current status of architecture work) show a certain degree of similarity with other studies on EA maturity. According to GAO (2002), IFEAD (2005) and NASCIO (2005), EA has been widely adopted by organizations, but the **EA maturity** seems to be on a quite initial level:

- Organizations may have defined their architectural frameworks and principles, and also the architectural guidance to ICT development projects is established to some extent.
- The actual architectural models are still generally under construction as well as the transition plan. The current state and the target state architectural models may exist on specific domains or viewpoints but are typically inconsistent and the big picture of the organizations's EA may still be fuzzy.
- Tool support is mainly limited to basic office tools and ICT development tools already introduced in organizations.
- Architectural evaluation and measurement is seen as an important issue but architectural work is seldom on such a high maturity level that evaluation is considered a useful activity in the organizations. Also the lack of evaluation practices is a challenge.
- Architecture work seems to be more a project level activity than a systematic enterprise-level approach in organizations.

Architecture evaluation (see the section on evaluation of architecture work) is a multifaceted instrument in architecture work. It seems that

- architecture evaluation is still more trigger-based than stabilized work in companies. A trigger may be, for example, a problem, a question or a need for information relating to the business or ICT-environment of the organization.
-
- architecture evaluation also has several meanings and roles in an organization and evaluations can be used for different purposes. The triggers revealed in our study (see the section on evaluation of architecture work) describe the role and meaning that architecture evaluation may have in an organization. In brief, architecture evaluations can be a tools for quality assurance, change management, architectural planning or IT cost management. Evaluations may also support the organizational planning and decision-making.
- different evaluation approaches are needed because architecture evaluation has different roles in different organizations.
- one challenge in architectural evaluations is the architectural documentation. Evaluation is typically based on documentation and descriptions. However, it is not clear or easy to decide what descriptions and documentation should be produced relating to architectures.
- there should be a relationship between architecture evaluations and organization's other measurement activities (such as balanced scorecards).

The **existing architecture evaluation methods** were also charted. The study

revealed that there seems to be a lack of methodologies evaluating EA. The most wide-spread approaches at the moment are maturity models and business-IT alignment assessment methods. Since no methods for the evaluation of the entire EA exist, techniques from the areas of business processes, data modelling, software architecture evaluation, benchmark testing, cost and benefits measurement of ICT investment were investigated (see the section on existing evaluation methods). Most of the introduced evaluation techniques are based on reviews of the architectural descriptions. Therefore, EA evaluation depends strongly on conceptual models as input and the basis for analysis and discussion because they support sharing and communicating the architectural knowledge among different stakeholders from different domains. It seems that a combination of methods is necessary to improve the fulfilment of certain EA evaluation needs. However, the complexity of EA and the related variety of concerns complicates reaching an established overall evaluation approach. So far it is only possible to apply different techniques to only single architectural views of EA.

While **architecture evaluation methods, metrics and criteria** are more established on the level of SA than EA, there was an attempt to clarify this area. EA evaluation is typically done as maturity evaluations and most of the existing methods and metrics support these activities. However, maturity evaluation is a quite superficial way to evaluate EA and may not be sufficient after the initial stages of EA development have passed. There is a definite need for more accurate and objective measures for different aspects of architecture and architecture work. We charted a vast amount of both qualitative and quantitative metrics for different areas of architectures and architecture work from literature, but the challenge of selecting the most suitable and useful for an organization remains (see the section on metrics and criteria).

In the complex and demanding business, IS development and software engineering context, the significance of well designed architectures and **high quality documentation** has been continually increasing. The challenges related to the architecture documentation in the organizations seems to be influenced by at least the following:

- multiple stakeholders of architecture work
- definition of the architecture framework and views used
- decisions concerning the documents to be produced
- multiple existing notations and tools and
- the lack of architecture documents, in some cases.

Architecture descriptions are used as communication tool. Architecture documents of bad quality may funnel the communication to irrelevant aspects. High quality documents enable more efficient architectural communication and enhance the understanding of the architecture. While the understanding of architecture can be seen as a prerequisite for the realization of architecture, the quality of architecture documents have, therefore, an effect on the realization of architectures. The quality of architecture documentation should be a concern of the architects, as well as of the whole company. We suggest that enterprise and software architects should ensure the quality of architecture documents while producing them. We also suggest that the quality check of architecture documents should be included in architecture reviews and quality evaluation checklists should be developed.

For the most part, at least in Finland companies are still initializing their EA efforts, and not so many architecture descriptions, models, or other artefacts exist. Hence, the evaluation of communication and common language, or commitment, is not considered to have the first priority. The problem with the evaluation of communication and common language is that the suggested metrics are to a large extent relative, or

subjective, trying to identify the level of satisfaction of a stakeholder. Also the evaluation of the benefits of the EA work is considered a challenging task.

In general, it seems that **metrics selection** is dependent on the phase of the architecture development, or more specifically, on the level of architecture maturity: simple metrics (e.g. on-off metrics) may be more usable in the beginning of the EA journey, and more detailed metrics (quantitative and qualitative metrics) may be utilized as the EA work is more established. Additionally, in different phases or maturity levels, different metrics are used. Most typically, simple metrics are needed in the initializing phase, and more advanced metrics (e.g. quantitative metrics) can be adopted in later phases.

In addition to several evaluation metrics defined for the above mentioned evaluation targets, we also studied the risks and decision-making in architecture work. An **overview of generic risks** that can potentially be related to EA in organizations was provided (see the section on architectural risks). EA risks were conceptualized both as factors that may lead to negative outcomes in the EA program, and as negative outcomes resulting from these factors. The risks were categorized using the work system framework. The results can be used to identify typical risks related to each element in the EA work system, and to assure that risk management practices have been planned for all relevant risks. Additionally, we suggest that EA risk management supports the attainment of EA objectives, or EA can even be exploited to facilitate organizational risk management.

Architecture decisions are high level decisions that, in the EA domain, can involve (see the section on decision-making)

- Selection of architecture plans (target, transition, vision)
- Selection of architecture standards, principles and guidelines
- Decisions about the objectives of architecture work in the organization

EA decisions are not necessarily official or actively made. The baseline architecture is constantly monitored and improvements planned but the big picture is not necessarily taken into account. The target architecture state are not necessarily officially approved, and the architecture transition plans may be working papers and the transitions are not necessarily systematic. Architectural decision making is dependent on the organization in question and

- decision makers involve various roles including architects, project roles and business management roles
- decision-making may be fragmented to various decision making points in the organizations
- EA team may have power over some decisions in the organization (e.g. IT portfolio and project planning) but may need approval for their own decisions from various points (e.g. business or IT management).

Finally, in practice, architectural work seems to be very different from theoretical frameworks and process models. There seems to be **a need for a light and agile EA methodology**, or at least a usable and simple enough EA process, in organizations initiating architectural work. The generic evaluation model for EA could be improved to be one possible tool to support organizations in launching the EA program. Similarly, the quality models developed in the project (see the section on quality management in the EA and SA context and the dissertation) provide a support for enhancing the software architecture work in organizations.

Top of page

Implications for Practitioners

Especially from the viewpoint of practitioners, the lessons learned are as follows.

Architectural work is a vast area, and **the success and quality of both EA and SA work seem to be influenced by multiple - and to some extent interrelated - factors** (see the section on architecture work). However, these factors can be used as a checklist by which practitioners both in the ICT user and service provider organizations undertaking, or planning to undertake, EA or SA efforts can ensure that the efforts are comprehensive, well-implemented, and have the minimum chance of failure. Additionally, the factors - the possible evaluation targets of architecture work - provide one usable starting point for the development of architecture evaluation criteria, metrics and methods.

Stakeholder identification is still difficult in practice for various reasons. Particularly, the EA stakeholders, their concerns and viewpoints are organization-specific at least to some extent, a certain stakeholder may fill several roles, and be a member of various stakeholder teams, groups or organizations, and the stakeholder viewpoints could be classified differently depending on the enterprise. To overcome these obstacles, the extensive framework including EA stakeholder roles, teams and organizations, and addressing their EA-related concerns and viewpoints, can be used in organizations to support identification of the stakeholders of EA and their concerns (see the section on stakeholders of architecture work).

Architects are dependent on the input and support of the various stakeholders. Therefore, architectural work involves **communication** to a great extent. Architects need to act as translators between stakeholders and constantly communicate the progress of the architecture program. The best way to communicate seems to be face-to-face. Information in a repository or in the corporate intranet supports communication but is not sufficient on its own.

The possible benefits of architecture work can be used as a basis for defining the objectives of architectural work in an enterprise. Additionally, architectural work may be rationalized, specifically to the management, in the initial stages by presenting the potential benefits which could be realized by architectural work. Benefits and their related metrics and evaluation criteria (see the section on evaluating the benefits of architecture work) can also be used as a basis for developing a measurement system for quantifying the value of architectural work.

The components of evaluation planning can be used in organizations to structure the planning phase of EA evaluation, and help to assure that all evaluation components are addressed before moving on to the actual evaluation. As a result, organizations could expect better comparability between the results of different evaluations, and greater results validity compared to an ad hoc approach. In addition, the given examples of EA objectives, evaluation purposes, audiences and evaluation targets can stimulate the discussion in organizations.

The results of the study on the evaluation of architecture documentation can be used in the producing these checklists. These checklists are suggested to be used in architecture design by architects and in architecture reviews by reviewers.

In addition, the wide selection of evaluation questions, criteria and metrics presented

related to communication, commitment, business-IT alignment and EA compliance (see the section on metrics) can be useful for organizations helping them define the few specific metrics for their needs.

Architectural risks are not necessarily taken into account in organizations. The overview and categorization of generic risks related to EA provided by the project (see the section on architectural risks) can be used to identify typical risks related to each element in the EA work system, and to assure that risk management practices have been planned for all relevant risks. Moreover, the EA work system framework may be used to structure the EA approach in organizations, regarding other aspects than risks as well.

Architectural decisions (see the section on architectural decision-making) should

- be made only if absolutely necessary to achieve business strategy and meet architectural objectives
- be traceable to business objectives
- not be overly detailed on the enterprise level
- take into account possible change needs to the architecture
- be enforceable and enforced
- be communicated with their rationale.

Architectural decision making is dependent on the organization in question; decision makers involve various roles including architects, project roles and business management roles and decision-making may be fragmented to various decision making points in the organizations. EA team may have power over some decisions in the organization (e.g. IT portfolio and project planning) but may need approval for their own decisions from various points (e.g. business or IT management). However, the following tips may support tackling the architectural decision-making in organizations:

- Plan architecture decision making and management:
    - define necessary decisions to enforce organizational strategies
    - define decision criteria
    - define how detailed should decisions be
    - define what kind of decisions should be officially approved
    - define where the decisions should be made and by whom
    - define who should gather the information required for decisions
    - define how are the decisions documented and communicated
    - define who enforces the decisions
- Cooperate with stakeholders in decision making; architecture may not have very established, official or influential position on its own.
- Communicate architecture decisions with their rationale to relevant stakeholders; merely storing decision documentation in a repository is not sufficient.

Future Trends

In the past, architecture has been considered more a technical approach but a change is evident. Architecture is quickly becoming an even more important **strategic management tool** for organizations in the future, as they seek to rationalize their operations and ICT portfolios as well as alignment between business and ICT. Even now EA development and governance are moving from the ICT department towards the

business. Similarly, the scope of EA will transform from the project level towards the enterprise level. Business planners and managers will apply EA for planning organizational improvements and gaining competitive advantage. Also enterprise architects will become more involved in the strategy and decision making processes in the organizations as they are they are able to understand both IT and business.

However, for EA to become such an extensively utilized management tool, there is a **need for better tool support**. EA stakeholders have different needs, different competencies, and use EA in different ways. EA tools should be able to present EA content in different ways to different stakeholders, and automatically support the consistency and completeness of architectural models. Such tools exist even now but they are not yet extensively spread. Organizations do not want to adopt a single tool, because it may not support the in-house frameworks and models, or may limit future improvements.

The **quality of architecture descriptions and documents** will also become a more important aspect. Descriptions, models and documents are essential in supporting architecture related communication and gaining a mutual understanding of the architecture visions, objectives, target states, and so forth. Therefore, for example, the modeling languages used in architecture modeling should be simple enough to be understandable and readable by the various stakeholder groups of the architecture work.

EA evaluation seems to be becoming an even more important activity for EA teams in the future (c.f. Liimatainen and Koskinen 2007; Rosen et al. 2007). As EA becomes a more mature, established practice in organizations, evaluation is required to assess the current status of EA, and to manage and improve it. In the future, especially the following evaluation targets will presumably become important:

- **EA benefits and value**: Displaying the realized value or benefits of EA will be important to enable EA teams to justify investments in EA. Even though it currently seems to be possible to justify EA investments with little solid evidence in many organizations (c.f. Aziz and Obitz 2007), this may not be the case in the future. Eventually, as the hype-value of EA diminishes and EA becomes a more established practice in organizations, the organizational management will probably require solid business cases for EA investments.

- **EA acceptance and utilization**: Evaluation of EA acceptance and use are important predictors of the success of the EA program – it is highly likely that no benefits are realized from EA if it is not actually applied in the organization. To become the stated strategic management tool, EA needs to be adopted as a tool in the day-to-day management decision-making. Property oriented EA evaluation techniques (see Winter et al. 2007) need to be applied to extract useful information from EA.

Also the evaluation practices will evolve and become more established. Evaluation and measurement of architecture work (process view) and architectures (product view) will become an integral part of architecture development and management. Especially in the SA domain, the importance of business requirements and needs will grow in the development and management of architectures.

All the above mentioned trends imply that the role and the profession of an enterprise architect will become perharps even more challenging. As Allen Brown puts it in the Architecture & Governance Magazine:

> " ... enterprise architects are now rated more highly than developers, when

> measured by the value they can deliver to their companies... It's worth noting that as the enterprise architecture profession continues to evolve and mature, there remains a shortage of qualified architects. Consequently, this demand is fueling the trend for hiring professionally certified enterprise architects." (Brown 2008)

Further Research

Which are the aspects or areas of EA and SA quality management and evaluation that were not considered during the research project? Which are the aspects or areas that still need to be scrutinized? To conclude this section, we provide the following suggestions for further research:

- Improving the **generic evaluation model for enterprise architecture**
    - Simplifying the model by combining areas (success factors) and modifying questions
    - Finding the most important areas for each level of maturity or in general
    - Finding the most important issues in each area
    - Defining simple and usable evaluation criteria and metrics to evaluate each area
    - Charting for new important issues not included in the model
    - Establishing distinct steps for moving to the next level of maturity
    - Studying the relationships or dependencies between the areas
- Constructing **evaluation methods and metrics for architecture benefits**
    - Defining the constructs that interact in the benefit realization process
    - Establishing causalities between constructs relating to architecture, architecture work and benefits
    - Displaying empirical evidence on architecture benefits
- Creating a **systematic, consistent architecture evaluation methodology**
    - Selecting and combining metrics and evaluation criteria
    - Selecting the most feasible metrics for each maturity level
    - Developing usable and effective processes or methods for evaluation
- Clarifying the **initialization phase of architecture work**
    - Creating agile or lightweight architecture development and management processes and practices for supporting systematic start-up of architecture work
    - Charting the available tool support for architecture work; how effective the tools are; can they really assist and ease the architecture work
    - Charting for best practices for establishing architecture work in an organization
- Studying the **implementation and utilization of architectures**
    - Charting for use cases of architecture and related practices
    - Finding ways to enhance architecture usage by stakeholders

- Charting for best practices for making architecture utilization and implementation a systematic, continuous process

Top of page Page Index

Bibliography

This section contains references related to the AISA project themes discussed in this CD-ROM publication. References deal with enterprise architecture, software architecture, quality management, measurement and evaluation of these architectures and architecture processes. In addition, a selection of architecture related web sites is provided.

Notice that the following lists may include links that are outdated.

- References
- A Selection of Enterprise Architecture Web Sites

See also the report describing a categorized listing of AISA project's bibliography and the bibliography in the publication Methods and Tools for Enterprise Architecture (Larkki project's documentation).

Top of page Page Index

Bibliography

AISA Project Report

Version: 1.0 Date: 13.8.2007
Author: Eetu Niemi, Tanja Ylimäki, Status: Final
Martin Hoffmann, Niina Hämäläinen

Contents

SUMMARY

This report lists and describes each of the categories of the literature found in the AISA project during the first and second years. The research subject of the AISA project is the quality management of enterprise and software architectures in the development of organizations and information systems, as well as related strategies, methods and tools. The project studies architectural key success factors, and evaluation criteria and metrics both at enterprise and software architecture level. In addition, the project investigates and develops quality management strategies and methods for architectures, particularly evaluation methods.

Main categories and subcategories listed and described in this bibliography are:

- architectural quality
  - quality and architecture,
  - enterprise architecture evaluation
  - software architecture evaluation
  - enterprise architecture success factors, evaluation criteria and metrics
  - software architecture success factors, evaluation criteria and metrics
- architecture evaluation methods
  - enterprise architecture evaluation methods
  - software architecture evaluation methods

In addition to these focus areas, the bibliography lists and describes some major references of relevant general background knowledge. Main categories and subcategories related to these areas listed in this bibliography are:

- architecture management background
  - enterprise architecture management
  - software architecture management
- quality and quality management background
  - quality
  - quality management
  - organizational quality and excellence
  - system quality, quality attributes and metrics

1 ARCHITECTURAL QUALITY

1.1 Quality and Architecture

The quality and architecture category includes all references which discuss the relationship between quality and architecture. Thus, it is a very large category including references related to studying architectural quality criteria, metrics and success factors, evaluation and analysis methods, and discussing the concept of architectural quality in general. The software architecture domain seems to be well represented in the literature, with some discussion on enterprise architecture quality, specifically maturity.

Abowd, G., L. Bass, P. Clements, R. Kazman, L. Northrop and A. Zaremski (1997). Recommended Best Industrial Practice for Software Architecture Evaluation, Technical Report CMU/SEI-96-TR-025, Software Engineering Institute (SEI), Carnegie Mellon University.

Al-Mashari, M. and M. Zairi (1999). "BPR implementation process: an analysis of key success and failure factors." Business Process Management Journal 5(1): 87-112.

Al-Naeem, T., I. Gorton, M. A. Babar, F. Rabhi and B. Benatallah (2005a). A quality-driven systematic approach for architecting distributed software applications. Proceedings of the 27th international conference on Software engineering. St. Louis, MO, USA, ACM Press.

Al-Naeem, T., I. Gorton, F. Rabhi and B. Benatallah (2005b). Tool support for optimization-based architectural evaluation. Proceedings of the second international workshop on Models and processes for the evaluation of off-the-shelf components. St. Louis, Missouri, ACM Press.

Avritzer, A. and E. J. Weyuker (1998). Investigating Metrics for Architectural Assessment. Proceedings of the Fifth International Software Metrics Symposium, Metrics 1998. Bethesda, MD, IEEE Computer Society: 4-10.

Avritzer, A. and E. J. Weyuker (1999). "Metrics to Assess the Likelihood of Project Success Based on Architecture Reviews." Empirical Software Engineering 4(3): 199 - 215.

Babar, M. A. and I. Gorton (2004). Comparison of Scenario-Based Software Architecture Evaluation Methods. Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSE'04), IEEE Computer Society: 600-607.

Babar, M. A., L. Zhu and R. Jeffery (2004). A Framework for Classifying and Comparing Software Architecture Evaluation Methods. Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), IEEE Computer Society.

Bachmann, F., L. Bass and M. Klein (2002). Illuminating the Fundamental Contributors to Software Architecture Quality, The Software Engineering Institute, Carnegie Mellon University.

Bachmann, F., L. Bass and M. Klein (2003). Deriving Architectural Tactics: A Step Toward Methodical Architectural Design, The Software Engineering Institute, Carnegie Mellon University.

Baker, D. C. and M. Janiszewski. (2005). "7 Essential Elements of EA." Enterprise Architect Retrieved 17.6.2005, from http://www.ftponline.com/ea/magazine/summer2005/features/dbaker/.

Barbacci, M., P. Clements, A. Lattanze, L. Northrop and W. Wood (2003a). Using the Architecture Tradeoff Analysis Method (ATAM) to Evaluate the Software Architecture for a Product Line of Avionics Systems: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M., R. Ellison, C. Weinstock and W. Wood (2000). Quality Attribute Workshop Participant's Handbook, Technical Report CMU/SEI-2000-SR-001, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M., M. H. Klein, T. A. Longstaff and C. B. Weinstock (1995). Quality Attributes, Technical Report CMU/SEI-95-TR-021, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R. (2002). SEI Architecture Analysis Techniques and When to Use Them, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., S. J. Carriere, P. H. Feiler, R. Kazman, M. H. Klein, H. F. Lipson, T. A. Longstaff and C. B. Weinstock (1997a). Steps in an Architecture Tradeoff Analysis Method: Quality Attribute Models and Analysis, Technical Report CMU/SEI-97-TR-029, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., R. Ellison, A. J. Lattanze, J. A. Stafford, C. B. Weinstock and W. G. Wood (2003b). Quality Attribute Workshops (QAWs), Third Edition, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., R. Ellison, J. A. Stafford, C. B. Weinstock and W. G. Wood (2001). Quality Attribute Workshops, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., M. H. Klein and C. B. Weinstock (1997b). Principles for Evaluating the Quality Attributes of a Software Architecture, Technical Report CMU/SEI-96-TR-036, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R. and W. G. Wood (1999). Architecture Tradeoff Analyses of C4ISR Products, The Software Engineering Institute, Carnegie Mellon University.

Barber, K. S. and J. Holt (2001). "Software Architecture Correctness." IEEE Software 18(6): 64-65.

Barber, K. S. G., T.J. (2000). Tool support for systematic class identification in object-oriented software architectures. Proceedings. 37th International Conference on Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Sydney, NSW.

Bass, L., P. Clements and R. Kazman (2003). Software Architecture in Practice, Addison-Wesley.

Bass, L. and B. E. John (2000). Achieving usability through software architectural styles. CHI '00 extended abstracts on Human factors in computing systems, The Hague, The Netherlands, ACM Press.

Bass, L. and B. E. John (2001). "Evaluating Software Architectures for Usability." Lecture Notes in Computer Science 2254.

Bass, L., B. E. John and J. Kates (2001a). Achieving Usability Through Software Architecture, The Software Engineering Institute, Carnegie Mellon University.

Bass, L., M. Klein and F. Bachmann (2000). Quality Attribute Design Primitives, Technical Note CMU/SEI-2000-TN-017, Software Engineering Institute, Carnegie Mellon University.

Bass, L., M. Klein and F. Bachmann (2002). Quality Attribute Design Primitives and the Attribute Driven Design Method. Lecture Notes in Computer Science. F. van der Linden, Springer-Verlag. 2290: 169-186.

Bass, L., M. Klein and G. Moreno (2001b). Applicability of General Scenarios to the Architecture Tradeoff Analysis Method, The Software Engineering Institute, Carnegie Mellon University.

Belle, J.-P. V. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. The 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries: 210-215.

Benarif, S., A. Ramdane-Cherif, N. Levy and F. Losavio (2004). Intelligent Tool Based-Agent for Software Architecture Evaluation. Proceedings of the Fourth International Conference on Quality Software (QSIC'04), IEEE Computer Society: 126-133.

Bengtsson, P. (1999). Design and Evaluation of Software Architecture. Department of Software Engineering and Computer Science. Karlskrona, University of Karlskrona/Ronneby.

Bengtsson, P. and J. Bosch (1998). Scenario-Based Software Architecture Reengineering. Proceedings of the 5th International Conference on Software Reuse. Victoria, BC, IEEE Computer Society: 308-317.

Bengtsson, P. and J. Bosch (1999). Architecture Level Prediction of Software Maintenance. Proceedings of the Third European Conference on Software Maintenance and Reengineering. Amsterdam, IEEE Computer Society: 139-147.

Bengtsson, P. and J. Bosch (1999). Haemo dialysis software architecture design experiences. Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, IEEE Computer Society Press.

Bergey, J., M. Barbacci and W. Wood (2000). Using Quality Attribute Workshops to Evaluate Architectural Design Approaches in a Major System Acquisition: A Case Study, Technical Note CMU/SEI-2000-TN-010, Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and P. C. Clements (2005a). Software Architecture in DoD Acquisition: A Reference Standard for a Software Architecture Document, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and P. C. Clements (2005b). Software Architecture in DoD Acquisition: An Approach and Language for a Software Development Plan, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and M. J. Fisher (2001). Use of the Architecture Tradeoff Analysis Method (ATAM) in the Acquisition of Software-Intensive Systems, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K., M. J. Fisher and L. G. Jones (2002). Use of the Architecture Tradeoff Analysis Method (ATAM) in Source Selection of Software-Intensive Systems, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K., M. J. Fisher, L. G. Jones and R. Kazman (1999). Software Architecture Evaluation with ATAM in the DoD System Acquisition Context, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and W. G. Wood (2002). Use of Quality Attribute Workshops (QAWs) in Source Selection for a DoD System Acquisition: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Bernus, P. (2003). "Enterprise Models for Enterprise Architecture and ISO9000:2000." Annual Reviews in Control 27: 211-220.

Boehm, B. (1994). Software architectures: critical success factors and cost drivers. Proceedings of the 16th International Conference on Software Engineering. Sorrento, Italy, IEEE Computer Society: 365.

Bosch, J. (1999). Product-line architectures in industry: a case study. Proceedings of the 21st international conference on Software engineering. Los Angeles, California, United States, IEEE Computer Society Press.

Bosch, J. and N. Juristo (2003). Designing software architectures for usability. Proceedings of the 25th International Conference on Software Engineering, Portland, Oregon, IEEE Computer Society.

Bosch, J. and P. Molin (1999). Software Architecture Design: Evaluation and Transformation. Proceedings of the IEEE Conference and Workshop on Engineering of Computer-Based Systems, ECBS '99. Nashville, TN, USA, IEEE Computer Society: 4-10.

Bosch, J. B., P. (2001). Assessing optimal software architecture maintainability. Fifth European Conference on Software Maintenance and Reengineering, 2001.

Bot, S., C.-H. Lung and M. Farrell (1996). A Stakeholder-Centric Software Architecture Analysis Approach. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, USA, ACM Press: 152-154.

Bowman, I. T. and R. C. Holt (1998). Software architecture recovery using Conway's law. Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, IBM Press.

Bratthall, L. and C. Wohlin (2002). "Is It Possible to Decorate Graphical Software Design and Architecture Models with Qualitative Information? - An experiment." IEEE Transactions on Software Engineering 28(12): 1181- 1193.

Bredemeyer Consulting. (2000). "Software Architecting Success Factors and Pitfalls." from http://www.bredemeyer.com/CSFs_pitfalls.htm.

Campbell, A., G. Coulson and D. Hutchison (1994). "A quality of service architecture." ACM SIGCOMM Computer Communication Review 24(2): 6 - 27.

Caporuscio, M., P. Inverardi and P. Pelliccione (2004). Compositional Verification of Middleware-Based Software Architecture Descriptions. Proceedings of the 26th International Conference on Software Engineering ICSE '04, IEEE Computer Society.

Carbone, J. (2004). "The Case for Architecture Assessment." Retrieved 7.4., 2005, from http://www.harriskern.com/index.php?m=p&authorid=567&pid=376&aid=155.

Carbone, J. (2005). "Ten Signs of Enterprise Architecture Maturity." 2005, from http://www.harriskern.com/index.php?m=p&pid=377&authorid=&aid=190.

Cheng, S.-W., R. L. Nord and J. A. Stafford (2005). "WICSA Wiki WAN Party: capturing experience in software architecture best practices." ACM SIGSOFT Software Engineering Notes 30(1): 1.

Choi, H. and K. Yeom (2002). An Approach to Software Architecture Evaluation with the 4+1 View Model of Architecture. Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02), IEEE Computer Society: 286-293.

Chrissis, M. B., M. Konrad and S. Shrum (2003). CMMI: Guidelines for Process Integration and Product Improvement, Addison Wesley.

Chung, L., B. A. Nixon and E. Yu (1995). An Approach to Building Quality into Software Architecture. Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research. Toronto, Ontario, Canada, IBM Press.

CIO Council (2000). Architecture Alignment and Assessment Guide, Chief Information Officers Council.

Clements, P., R. Kazman and M. Klein (2001). Evaluating Software Architectures: Methods and Case Studies. Boston, USA, Addison-Wesley.

Clements, P. C. (1995). Understanding Architectural Influences and Decisions in Large-System Projects. First International Workshop on Architectures for Sofware Systems, Seattle.

Cockburn, A. (1996). "The Interaction of Social Issues and Software Architecture." Communications of the ACM 39(10): 40-46.

Corry, A. V., J. Bardram, H. B. Christensen, M. Ingstrup and K. M. Hansen (2005). Exploring quality attributes using architectural prototyping. Proceedings of the First International Conference on the Quality of Software Architectures (QoSA 2005), Springer-Verlag, Berlin Heidelberg.

Crispen, R. G. and L. D. Stuckey (1994). Structural model: architecture for software designers. Proceedings of the conference on TRI-Ada '94. Baltimore, Maryland, United States, ACM Press.

Curran, C. (2005). "Link IT Investments to Business Metrics." Enterprise Architect 3(1): 16-18.

de Bruin, H. and H. van Vliet (2003). "Quality-Driven Software Architecture Composition." Journal of Systems and Software 66(3): 269-284.

DeLone, W. and E. McLean (1992). "Information Systems Success: The Quest for the Dependent Variable." Information Systems Reseach 3(1): 60-95.

Dias, M. S. and M. E. R. Vieira (2000). Software Architecture Analysis Based on Statechart Semantics. Proceedings of the Tenth International Workshop on Software Specification and Design. San Diego, CA, IEEE Computer Society: 133-137.

Dias, O. P., I. C. Teixeira and J. P. Teixeira (1999). "Metrics and Criteria for Quality Assessment of Testable Hw/Sw Systems Architectures." Journal of Electronic Testing: Theory and Applications 14: 149-158.

Díaz-Pace, J. A. and M. R. Campo (2005). ArchMatE: from architectural styles to object-oriented models through exploratory tool support. Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications. San Diego, CA, USA, ACM Press.

Dikel, D., D. Kane, S. Ornburn and J. Wilson. (1995). "Software Architecture Case Study: Organizational Success Factors." from http://www.vraps.com/files/archcase_exec_sum.doc http://www.vraps.com/files/.

Dobrica, L. and E. Niemelä (2002). "A Survey on Software Architecture Analysis Methods." IEEE Transactions on Software Engineering 28(7): 638-653.

DoC (2003). IT Architecture Capability Maturity Model. Department of Commerce (DoC).

Dolan, T., J. (2001). Architecture Assessment of Information-System Families. Eindhoven, The Netherlands, Technische Universiteit Eindhoven: 222.

Dueñas, J. C., W. L. d. Oliveira and J. A. d. l. Puente (1998). "A Software Architecture Evaluation Model." Lecture Notes in Computer Science 1429: 148-157.

Eguiluz, H. R. and M. R. Barbacci (2003). Interactions Among Techniques Addressing Quality Attributes, The Software Engineering Institute, Carnegie Mellon University.

Ellison, R. J., A. P. Moore, L. Bass, M. Klein and F. Bachmann (2004). Security and Survivability Reasoning Frameworks and Architectural Design Tactics, The Software Engineering Institute, Carnegie Mellon University.

Erder, M. and P. Pureur (2003). "QFD in the Architecture Development Process." IT Professional 5(6): 44-52.

ETS Office (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

Everitt, T., R. Tvedt, T. and J. Tvedt, D. (2004). Validating and Improving an Existing Software Architectural Evaluation Process. Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04): 417-421.

F. Lasavio, L. C. (2004). "ISO quality standards for measuring architectures." The Journal of Systems and Software 72: 209 - 223.

Fabbrini, F., M. Fusani and S. Gnesi (1998). Quality Evaluation based on Architecture Analysis. Proceedings of the International Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA'98). Marsala, Italy.

Folmer, E. and J. Bosch (2004). "Architecting for Usability: a Survey." The Journal of Systems and Software 70: 61-78.

Folmer, E., J. van Gurp and J. Bosch (2003). Scenario-Based Assessment of Software Architecture Usability. Proceedings of the International Conference on Software Engineering (ICSE 2003). Portland, Oregon, USA: 61-68.

Fraser, P., J. Moultrie and M. Gregory (2002). The use of maturity models/grids as a tool in assessing product development capability. IEEE International Engineering Management Conference. Cambridge, IEEE.

Fresa, A., G. Nucera, E. Peciola and G. Santucci (2002). Assessment of software architectures: a case study. Proceedings of the 14th international conference on Software engineering and knowledge engineering SEKE '02, Ischia, Italy, ACM Press.

Fukuzawa, K. and M. Saeki (2002). Evaluating Software Architectures by Coloured Petri Nets. Proceedings of the 14th international conference on Software engineering and knowledge engineering. Ischia, Italy, ACM Press: 263-270.

Gallagher, B. P. (2000). Using the Architecture Tradeoff Analysis Method to Evaluate a Reference Architecture: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Gannod, G. C. and R. R. Lutz (2000). An approach to architectural analysis of product lines. Proceedings of the 22nd international conference on Software engineering. Limerick, Ireland, ACM Press.

GAO (2002). Enterprise Architecture Use across the Federal Government Can Be Improved. USA: Washington, D.C., United States General Accounting Office, USA.

GAO. (2003a). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." from http://www.gao.gov/new.items/d03584g.pdf.

GAO (2003b). Leadership Remains Key to Agencies Making Progress on Enterprise Architecture Efforts, General Accounting Office (GAO).

Garlan, D., R. Allen and J. Ockerbloom (1995). Architectural Mismatch or Why It's Hard to Build Systems out of Existing Parts. Proceedings of the 17th international conference on Software engineering. Seattle, Washington, United States, ACM Press: 179-185.

Golden, E. J., B.E.; Bass, L. (2005). Quality vs. quantity: comparing evaluation methods in a usability-focused software architecture modification task. International Symposium on Empirical Software Engineering.

Gomaa, H. (2004). Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures, Addison Wesley.

Goseva-Popstojanova, K., A. Hassan, G. A., W. Abdelmoez, D. E. M. Nassar, H. Ammar and A. Mili (2003). "Architectural-level risk analysis using UML." IEEE Transactions on Software Engineering 29(10): 946 - 960.

Government Accountability Office (GAO). (2003). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." 2004.

Gustafsson, J., J. Paakki, L. Nenonen and A. I. Verkamo (2002). Architecture-Centric Software Evolution by Software Metrics and Design Patterns. Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02). Budapest, IEEE Computer Society: 108-115.

Hai-Shan, C. (2004). Survey on the style and description of software architecture. Proceedings. The 8th International Conference on Computer Supported Cooperative Work in Design, 2004.

Haley, G. (2005). "Software Architecture Evaluation - Transforming a craft into a business process." Retrieved 26.5., 2005, from http://sunset.usc.edu/gsaw/gsaw2005/s9b/haley.pdf.

Hazra, T. K. (2005). "Getting Your EA Metrics Right (a presentation)." 2005.

Hilliard, R., M. Kurland, J., S. Litvintchouk, D., T. Rice and S. Schwarm (1996). Architecture Quality Assessment, version 2.0, The MITRE Corporation.

Hilliard, R., M. J. Kurland and S. D. Litvintchouk (1997). MITRE's Architecture Quality Assessment. Proceedings of the Software Engineering & Economics Conference.

Hirvonen, A. and M. Pulkkinen (2003). Evaluation of IT Architecture Solutions - How Can an ICT Consultant Tell What Is Best for You? Proceeding of the 10th European Conference on Information Technology Evaluation, Management Centre International Limited.

Hämäläinen, N. (2005). Quality Management Activities in Software Architecture Management (manuscript).

Ifinedo, P. E. (2006). Enterprise Resource Planning Systems Success Assessment: An Integrative Framework. Department of Computer Science and Information Systems. Jyväskylä, FInland, University of Jyväskylä. PhD: 153.

Ionita, M. T., P. America, D. K. Hammer, H. Obbink and J. J. M. Trienekens (2004). A scenario-driven approach for value, risk, and cost analysis in system architecting for innovation. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Ionita, M. T., D. K. Hammer and H. Obbink (2002). Scenario-Based Software Architecture Evaluation Methods: An Overview. Proceedings of the International Conference on Software Engineering (ICSE 2002). Orlando, Florida, USA.

Issarny, V., C. Kloukinas and A. Zarras (2002). "Systematic aid for developing middleware architectures." Communications of the ACM 45(6): 53 - 58.

Issarny, V., T. Saridakis and A. Zarras (1998). Multi-view description of software architectures. Proceedings of the third international workshop on Software architecture, Orlando, Florida, United States, ACM Press.

Ivkovic, I. K., K. (2006). A Framework for Software Architecture Refactoring using Model Transformations and Semantic Annotations. Proceedings of the 10th European Conference on Software Maintenance and Reengineering, CSMR 2006.

Jones, L. G. and A. J. Lattanze (2001). Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Kamogawa, T. and H. Okada (2005). A Framework for Enterprise Architecture Effectiveness. Proceedings of the Second International Conference on Services Systems and Services Management (ICSSSM '05). Chongqing, China, IEEE Computer Society.

Kavi, K. M. and K. Krishnamohan (1984). "Architecture Quality." ACM SIGOPS Operating Systems Review 18(1): 11-19.

Kazman, R. (1996). Tool Support for Architecture Analysis and Design. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, California, United States: 94-97.

Kazman, R., G. Abowd, L. Bass and P. Clements (1996). "Scenario-Based Analysis of Software Architecture." IEEE Software 13(6): 47-55.

Kazman, R., J. Asundi and M. Klein (2001). Quantifying the Costs and Benefits of Architectural Decisions. Proceedings of the 23rd International Conference on Software Engineering. Toronto, Ontario, Canada, IEE Computer Society: 297-306.

Kazman, R., M. Barbacci, M. Klein, S. J. Carrière and S. G. Woods (1999). Experience with Performing Architecture Tradeoff Analysis. Proceedings of the 21st international conference

on Software engineering. Los Angeles, California, United States, IEEE Computer Society Press: 54-63.

Kazman, R. and L. Bass (2002). "Making Architecture Reviews Work in the Real World." IEEE Software 19(1): 67-73.

Kazman, R., L. Bass, G. Abowd and M. Webb (1994). SAAM: A Method for Analyzing the Properties of Software Architectures. Proceedings of the 16th International Conference on Software Engineering: 81-90.

Kazman, R., L. Bass, G. Abowd and M. Webb (1998a). SAAM: A Method for Analyzing the Properties of Software Architectures. Proceedings of the 16th International Conference on Software Engineering: 81-90.

Kazman, R. and S. J. Carriere (1997). Playing Detective: Reconstructing Software Architecture from Available Evidence, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R. and M. Klein (1998). Performing Architecture Tradeoff Analysis. Proceedings of the third international workshop on software architecture. Orlando, Florida, USA, ACM Press: 85-88.

Kazman, R., M. Klein, M. Barbacci, T. Longstaff, H. Lipson and J. Carriere (1998b). The Architecture Tradeoff Analysis Method, Technical Report CMU/SEI-98-TR-008, Software Engineering Institute, Carnegie Mellon University.

Kazman, R., M. Klein and P. Clements (2000). ATAM: Method for Architecture Evaluation, Technical Report CMU/SEI-2000-TR-004, Software Engineering Institute, Carnegie Mellon University.

Kazman, R., P. Kruchten, R. L. Nord and J. E. Tomayko (2004). Integrating Software-Architecture-Centric Methods into the Rational Unified Process, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R., R. L. Nord and M. Klein (2003). A Life-Cycle View of Architecture Analysis and Design Methods, The Software Engineering Institute, Carnegie Mellon University.

Khajenoori, S., L. Prem, K. Stevens, B. S. Keng and N. Kameli (2004). "Knowledge Centered Assessment Pattern: an Effective Tool for Assessing Safety Concerns in Software Architecture." The Journal of Systems and Software 73: 313-322.

Kishi, T. and N. Noda (2001). Aspect-oriented analysis for architectural design. Proceedings of the 4th International Workshop on Principles of Software Evolution. Vienna, Austria, ACM Press.

Kluge, C., A. Dietzsch and M. Rosemann (2006). How to Realize Corporate Value from Enterprise Architecture. the Proceedings of the 14th European Conference on Information Systems (ECIS 2006). Göteborg, Sweden, Association for Information Systems.

Knodel, J. L., M.; Muthig, D. (2005). Static Evaluation of Software Architectures - A Short Summary. 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005.

Krikhaar, R. P., A.; Sellink, A.; Stroucken, M.; Verhoef, C. (1999). A two-phase process for software architecture improvement. Proceedings. IEEE International Conference on Software Maintenance, 1999. (ICSM '99).

Krueger, I. H. (2004). Evaluating Software Architectures: Stakeholders, Metrics, Results, Migration Strategies. Department of Computer Science & Engineering. San Diego, University of California.

Kulpa, M. K. and K. A. Johnson (2003). Interpreting the CMMI: A Process Improvement Approach, Auerbach Publications.

Lassing, N., P. Bengtsson, H. van Vliet and J. Bosch (2002). "Experiences with ALMA: Architecture-Level Modifiability Analysis." Journal of Systems and Software 61(1): 47-57.

Lassing, N., D. Rijsenbrij and H. van Vliet (1999a). The Goal of Software Architecture Analysis: Confidence Building or Risk Assessment. Proceedings of the First Benelux Conference on State-of-the-Art of ICT Architecture.

Lassing, N., D. Rijsenbrij and H. van Vliet (1999b). On Software Architecture Analysis of Flexibility; Complexity of Changes: Size isn't Everything. Proceedings of the Second Nordic Workshop on Software Architecture (NOSA99).

Lassing, N., D. Rijsenbrij and H. van Vliet (1999c). Towards a Broader View on Software Architecture Analysis of Flexibility. Proceedings of the Sixth Asia Pacific Software Engineering Conference (APSEC '99). Takamatsu, IEEE Computer Society: 238-245.

Lassing, N., D. Rijsenbrij and H. van Vliet (2001). "Viewpoints on Modifiability." International Journal of Software Engineering and Knowledge Engineering 11(4): 453-478.

Lehto, J., A. and P. Marttiin (2005). Experiences in System Architecture Evaluation: A Communication View for Architectural Design. Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS'05), IEEE Computer Society: 312c-312c.

Leonard, M. S. and J. B. Bowles (1994). The Application of QFD to Computer-System Architecture. Proceedings of the Annual Symposium on Reliability and Maintainability. Anaheim, CA, IEEE Computer Society: 359.

Liang, S. X. and V. B. Luqi (2003). "Quantifiable architecting of dependable systems of embedded systems." ACM SIGSOFT Software Engineering Notes 28(6): 7 - 7.

Lindvall, M., R. Tesoriero and P. Costa (2002). Avoiding Architectural Degeneration: An Evaluation Process for Software Architecture. Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS'02), IEEE Computer Society: 77-86.

Lopez, M. (2000). An Evaluation Theory Perspective of the Architecture Tradeoff Analysis Method (ATAM). Pittsburg, USA, The Software Engineering Institute, Carnegie Mellon University.

Losavio, F., L. Chirinos, N. Lévy and A. Ramdane-Cherif (2003). "Quality Characteristics for Software Architecture." Journal of Object Technology 2(2): 133-150.

Losavio, F., L. Chirinos, A. Matteo, N. Lévy and A. Ramdane-Cherif (2004). "ISO Quality Standards for Measuring Architectures." The Journal of Systems and Software 72: 209-223.

Losavio, F., A. Matteo, J. O. Ordaz, N. Levy and R. Marcano-Kamenoff (2002). Quality Characteristics to Select an Architecture for Real-time Internet Applications. 4th International Software Quality Week Europe (QWE2000). Brussels, Belgium.

Luftman, J. (2000). "Assessing Business-IT Alignment Maturity." Communications of the Association for Information Systems 4(Article 14).

Luftman, J. N., R. Papp and T. Brier (1999). "Enablers and Inhibitors of Business-IT Alignment." Communications of the Association for Information Systems 1(11).

Lundberg, L., J. Bosch, D. Häggander and P. Bengtsson (1999). Quality Attributes in Software Architecture Design. Proceedings of the IASTED 3rd International Conference on Software Engineering and Applications.

Lung, C.-H., S. Bot, K. Kalaichelvan and R. Kazman (1997). An Approach to Software Architecture Analysis for Evolution and Reusability. Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research. Toronto, Ontario, Canada, IBM Press: 15-26.

Lung, C.-H., A. Jalnapurkar and A. El-Rayess (1998). Performance-oriented software architecture engineering: an experience report. Proceedings of the 1st international workshop on Software and performance WOSP '98, Santa Fe, New Mexico, United States, ACM Press.

Lung, C.-H. and K. Kalaichelvan (2000). "An Approach to Quantitative Software Architecture Sensitivity Analysis." International Journal of Software Engineering and Knowledge Engineering 10(1): 97-114.

Ma, R. (1996). "Quality System an Integral Part of Total Quality Management." Computers & Industrial Engineering 31(3/4): 753-757.

Maccari, A. (2002). Experiences in Assessing Product Family Software Architecture for Evolution. Proceedings of the 24th International Conference on Software Engineering. Orlando, Florida, USA, ACM Press: 585-592.

Magee, S. and D. Thiele (2004). "Engineering Process Standards: State of the Art and Challenges." IT Professional 6(5): 38-44.

Malek, S., M. Mikic-Rakic and N. Medvidovic (2004). An extensible framework for autonomic analysis and improvement of distributed deployment architectures. Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems WOSS '04. Newport Beach, California, ACM Press.

Maranzano, J. F., S. A. Rozsypal, G. H. Zimmerman, G. W. Warnken, P. E. Wirth and D. M. Weiss (2005). "Architecture Reviews: Practice and Experience." IEEE Software 22(2): 34-43.

Matinlassi, M. (2004). Evaluating the portability and maintainability of software product family architecture: terminal software case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Matinlassi, M., E. Niemelä and L. Dobrica (2002). Quality-Driven Architecture Design and Quality Analysis Method - A Revolutionary Initiation Approach to a Product Line Architecture. VTT Publications. Espoo, Technical Research Centre of Finland, VTT: 128.

McCay, B. M. (1996). Some Thoughts on the Quality of a Computer-Based System's Architecture. Engineering of the IEEE Symposium and Workshop on Computer-Based Systems. Friedrichshafen, IEEE Computer Society: 228-234.

McGovern, J., S. W. Ambler, M. Stevens, J. Linn, V. Sharan and E. K. Jo (2003). A Practical Guide to Enterprise Architecture. New Jersey, Prentice Hall PTR.

META Group Inc. (2000a). "Architecture Capability Assessment." META Practice 4(7).

META Group Inc. (2000b). "Architecture Maturity Assessment." METAView Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0378/mv0378.html.

META Group Inc. (2002). "Enterprise Architecture Maturity." METAView, Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0599/mv0599.html.

META Group Inc. (2004a). "Architecture Program Maturity Assessment: Findings and Trends." Enterprise Planning & Architecture Strategies, Teleconference 2118, 23 September 2004, from http://www.metagroup.com/us/displayArticle.do?oid=49449.

META Group Inc. (2004b). "Planning the Enterprise Architecture Measurement Program." META Group Research, Enterprise Planning & Architecture Strategies, Practice Summary, from http://www.metagroup.com/us/displayArticle.do?oid=50037.

Moore, M., R. Kazman, M. Klein and J. Asundi (2003). Quantifying the Value of Architecture Design Decisions: Lessons from the Field. Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society: 557-562.

Morganwalp, J. and A. P. Sage (2003). "A System of Systems Focused Enterprise Architecture Framework and an Associated Architecture Development Process." Information Knowledge Systems Management 3(2): 87-105.

Morganwalp, J. M. and A. P. Sage (2004). "Enterprise Architecture Measures of Effectiveness." International Journal of Technology, Policy and Management 4(1): 81-94.

Morisio, M., I. Stamelos and A. Tsoukias (2002). A new method to evaluate software artifacts against predefined profiles. Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE 2002). Ischia, Italy, ACM Press: 811-818.

Muccini, H. D., M.; Richardson, D.J. (2005). Reasoning about software architecture-based regression testing through a case study. 29th Annual International Computer Software and Applications Conference, 2005. COMPSAC 2005.

NASCIO (2003). NASCIO Enterprise Architecture Maturity Model, v. 1.3, National Association of State Chief Information Officers (NASCIO).

NASCIO. (2004). "NASCIO Enterprise Architecture Maturity Model, v. 3.1 Self-Assessment (Draft)." 2004, from https://www.nascio.org/publications/index.cfm.

NASCIO (2005). The States and Enterprise Architecture: How far have we come? Findings from the NASCIO 2005 EA Assessment. Lexington, National Association of State Chief Information Officers (NASCIO), USA.

Noda, N. and T. Kishi (1999). On Aspect-Oriented Design - An Approach to Designing Quality Attributes. Proceedings of the Sixth Asia Pacific Software Engineering Conference (APSEC '99). Takamatsu, Japan: 230-237.

Nord, R. L., M. R. Barbacci, P. Clements, R. Kazman, M. Klein, L. O'Brien and J. E. Tomayko (2003). Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM), The Software Engineering Institute, Carnegie Mellon University.

OMB. (2004). "Guidelines for Enterprise Architecture Assessment Framework."

OMB (2005a). Federal Enterprise Architecture Program EA Assessment Framework 2.0, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB (2005b). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB FEA Program Management Office. (2004). "Guidelines for Enterprise Architecture Assessment Framework."

OMB FEA Program Management Office (2005). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

Paoli, F. D. and A. Sosio (1996). Requirements for a layered software architecture supporting cooperative multi-user interaction. Proceedings of the 18th international conference on Software engineering, Berlin, Germany, IEEE Computer Society.

Peters, P., P. Szczurko, M. Jarke and M. Jeusfeld (1995). Business process oriented information management: conceptual models at work. Proceedings of conference on Organizational computing systems. Milpitas, California, United States, ACM Press.

Petriu, D. S., C.; Jalnapurkar, A. (2000). "Architecture-based performance analysis applied to a telecommunication system." IEEE Transactions on Software Engineering 26(11): 1049 - 1065.

Piattini, M. C., C.; Astudillo, H. (2005). Classifying Software Architecture Quality Research. 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005.

Purhonen, A. (2004). Performance optimization of embedded software architecture - a case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Quality Assurance Project. (2006). "A Glossary of Useful Terms." Retrieved 11 November, 2006.

Rehkopf, T. W. and N. Wybolt (2003). "Top 10 Architecture Land Mines." IT Professional 5(6): 36-43.

Richardson, D. J. and A. L. Wolf (1996). Software Testing at the Architectural Level. Joint Proceedings of the Second International Software Architecture Workshop (ISAW-2) and International Workshop on Multiple Perspectives in Software Development (Viewpoints '96) on SIGSOFT '96 Workshops. San Francisco, California, United States, ACM Press: 68-71.

Rico, D. F. (2005). "A Framework for Measuring the ROI of Enterprise Architecture."

Rosenfeld, L., K. Instone, S. Cupala, J. J. Garrett, M. Hearst, G. Marchionini and N. Ragouzis (2001). Measuring Information Architecture Quality: Prove It (or not)! CHI '01 extended abstracts on Human factors in computing systems. Seattle, Washington, USA, ACM Press: 219-220.

Saleh, Y. and M. Alshawi (2005). "An alternative model for measuring the success of IS projects: the GPIS model." Journal of Enterprise Information Management 18(1): 47-63.

Scallon, G. (1999). "Model Based Development." Retrieved 26.5.2005, from http://www.stc-online.org/cd-rom/1999/slides/MBDslid1.pdf.

Schekkerman, J. (2003). "Extended Enterprise Maturity Model (E2AMM)." from http://www.enterprise-architecture.info/Images/E2AF/E2AMMv2.PDF.

Schekkerman, J. (2004a). Enterprise Architecture Validation - Achieving Business-Aligned and Validated Enterprise Architectures, Institute For Enterprise Architecture Developments, URL: http://www.enterprise-architecture.info/: 27.

Schekkerman, J. (2004b). How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework. Canada, Trafford.

Schekkerman, J. (2004a). Enterprise Architecture Validation - Achieving Business-Aligned and Validated Enterprise Architectures, Institute For Enterprise Architecture Developments, URL: http://www.enterprise-architecture.info/: 27.

Schekkerman, J. (2004b). How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework. Canada, Trafford.

Schekkerman, J. (2005). "Trends in Enterprise Architecture 2005 - How are Organizations Progressing? Web-form Based Survey 2005." Retrieved August 15, 2006, from http://www.enterprise-architecture.info/Images/EA%20Survey/Enterprise%20Architecture%20Survey%202005%20IFEAD%20v10.pdf.

Schmerl, B. and D. Garlan (2004). AcmeStudio: Supporting Style-Centered Architecture Development. Proceedings of the 26th International Conference on Software Engineering, IEEE Computer Society.

Sharareh, A., G. Matteo and T. Gianluca (2002). Quantitative analysys for telecom/datacom software architecture. Proceedings of the 3rd international workshop on Software and performance WOSP '02, Rome, Italy, ACM Press.

Shereshevsky, M., H. Ammari, N. Gradetsky, A. Mili and H. H. Ammar (2001). Information Theoretic Metrics for Software Architectures. Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC 2001), Chicago, IL, USA, IEEE Computer Society.

Simone, M. D. and R. Kazman (1995). Software architectural analysis: an experience report. Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, IBM Press.

Sullivan, K. J. and J. C. Knight (1996). Experience assessing an architectural approach to large-scale systematic reuse. Proceedings of the 18th international conference on Software engineering. Berlin, Germany, IEEE Computer Society.

Svahnberg, M. (2004). "An Industrial Study on Building Consensus Around Software Architectures and Quality Attributes." Information and Software Technology 46: 805-818.

Svahnberg, M. and C. Wohlin (2002). Consensus Building when Comparing Software Architectures. Proceedings of the 14th International Conference on Product Focused Software Process Improvement (PROFES 2002), Lecture Notes in Computer Science (LNCS). M. Oivo and S. Komi-Sirviö. Rovaniemi, Finland, Springer. 2559: 436-452.

Svahnberg, M. and C. Wohlin (2005). "An Investigation of a Method for Identifying a Software Architecture Candidate with Respect to Quality Attributes." Empirical Software Engineering 10: 149-181.

Svahnberg, M., C. Wohlin, L. Lundberg and M. Mattsson (2002). A Method for Understanding Quality Attributes in Software Architecture Structures. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering. Ischia, Italy, ACM Press: 819-826.

Tahvildari, L., K. Kontogiannis and J. Mylopoulos (2003). "Quality-Driven Software Re-engineering." The Journal of Systems and Software 66: 225-239.

Taylor, R. N., W. Tracz and L. Coglianese (1995). "Software development using domain-specific software architectures: CDRl A011—a curriculum module in the SEI style." ACM SIGSOFT Software Engineering Notes 20(5): 27 - 38.

Tekinerdogan, B. (2004). ASAAM: aspectual software architecture analysis method. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Teo, T. S. H. and J. S. K. Ang (1999). "Critical success factors in the alignment of IS plans with business plans." International Journal of Information Management 19(2): 173-185.

The Office of Enterprise Technology Strategies (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

Tvedt, R. T., M. Lindvall and P. Costa (2002). A Process for Software Architecture Evaluation Using Metrics. Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02): 191-196.

Van Belle, J. P. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. Proceedings of the 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (SAICSIT-2004): 210-215.

van den Bent, B. (2006). A Quality Intstrument for the Enterprise Architecture Development Process. Institute of Information and Computing Sciences. Utrecht, Utrecht University: 78.

van der Raadt, B., J. Hoorn, F. and H. van Vliet (2005). Alignment and Maturity are Siblings in Architecture Assessment. Proceedings of the 17th Conference on Advanced Information Systems Engineering, CAiSE 2005. Porto, Portugal, Springer.

van der Raadt, B., J. Soetendal, M. Perdeck and H. v. Vliet (2004). Polyphony in Architecture. Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), IEEE Computer Society: 533-542.

Wang, G. C., A.; Wang, C.; Fung, C.; Uczekaj, S. (2004). Integrated quality of service (QoS) management in service-oriented enterprise architectures. Proceedings. Eighth IEEE International Enterprise Distributed Object Computing Conference, EDOC 2004.

Wang, T., A. Hassan, A. Guedem, W. Abdelmoez, K. Goseva-Popstojanova and H. Ammar (2003). Architectural level risk assessment tool based on UML specifications. Proceedings of the 25th International Conference on Software Engineering. Portland, Oregon, IEEE Computer Society.

Veltman-van Reekum, E. (2006). Determining the Quality of Enterprise Architecture Products. Institute of Information and Computing Sciences. Utrecht, Utrecht University: 110.

Weyuker, E. J. (1999). "Evaluation techniques for improving the quality of very large software systems in a cost-effective way." The Journal of Systems and Software 47: 997-103.

Vieira, M. E. R., Marcio S. Dias and D. J. Richardson (2000). Analyzing software architectures with Argus-I. Proceedings of the 22nd international conference on Software engineering, Limerick, Ireland, ACM Press.

Vigder, M., T. McClean and F. Bordeleau (2003). "Evaluating COTS Based Architectures." Lecture Notes in Computer Science 2580: 240 - 250.

Wijnstra, J. G. (2003). "From problem to solution with quality attributes and design aspects." The Journal of Systems and Software 66: 199-211.

Wilkin, C. and B. Hewett (1999). Quality in a Respecification of DeLone and McLean's IS Success Model. Managing Information Technology Resources in Organizations in the Next Millennium. M. Khosrowpour. Hershey, USA, Idea Group Publishing: 663-672.

Williams, L. G. and C. U. Smith (1998). Performance evaluation of software architectures. Proceedings of the 1st international workshop on Software and performance WOSP '98, Santa Fe, New Mexico, United States, ACM Press.

Williams, L. G. and C. U. Smith (2002). PASA: A Method for the Performance Assessment of Software Architectures. Proceedings of the 3rd Workshop on Software Performance, WOSP '02. Rome, Italy: 179-189.

Woods, S. G. and M. Barbacci (1999). Architectural Evaluation of Collaborative Agent-Based Systems, The Software Engineering Institute, Carnegie Mellon University.

Woody, C. (2005). Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements, The Software Engineering Institute, Carnegie Mellon University.

Wu, C. C., E. (2005). Comparison of Web service architectures based on architecture quality properties. 3rd IEEE International Conference on Industrial Informatics, INDIN '05.

Yakimovich, D., J. M. Bieman and V. R. Basili (1999). Software architecture classification for estimating the cost of COTS integration. Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, IEEE Computer Society Press.

Zarayaraz, G., P. Thambidurai, S. Kappuswami and P. Rodrigues (2003). "New Approach to SW Architecture." SIGSOFT Software Engineering Notes 28(2): 17.

Zayaraz, G., D. P. Thambidurai, M. Srinivasan and D. P. Rodrigues (2005). "Software quality assurance through COSMIC FFP." ACM SIGSOFT Software Engineering Notes 30(5): 1 - 5.

Zayaraz, G. T., P. (2005). "Software Architecture Selection Framework Based on Quality Attributes." Annual IEEE INDICON, 2005: 167 - 170.

Zeng, D. D. and J. L. Zhao (2002). Achieving Software Flexibility via Intelligent Workflow Techniques. Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02), IEEE Computer Society: 606-615.

Zhu, L., M. A. Babar and R. Jeffery (2004). Mining Patterns to Support Software Architecture Evaluation. Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), IEEE Computer Society: 25-34.

Zijden, S., H. Goedvolk and D. Rijsenbrij. (2000). "Architecture: Enabling Business and IT Alignment in Information System Development." Cap Gemini White Paper, from http://home.hetnet.nl/~daan.rijsenbrij/arch/publ/artarc05.doc.

1.2 Enterprise Architecture Evaluation

This category lists references discussing enterprise architecture evaluation in general. Some references may include metrics or quality criteria, but mainly the focus in on evaluation approaches and methods. Especially, focus is on enterprise architecture maturity evaluation approaches developed in the US government. Moreover, enterprise architecture surveys conducted in the field are included in this category.

Al-Mashari, M. and M. Zairi (1999). "BPR implementation process: an analysis of key success and failure factors." Business Process Management Journal 5(1): 87-112.

Aziz, S., T. Obitz, R. Modi and S. Sarkar. (2006). "Enterprise Architecture: A Governance Framework - Part II: Making Enterprise Architecture Work within the Organization." Infosys White Paper Retrieved 22 August, 2006, from http://www.infosys.com/enterprise-architecture/.

Baker, D. C. and M. Janiszewski. (2005). "7 Essential Elements of EA." Enterprise Architect Retrieved 17.6.2005, from http://www.ftponline.com/ea/magazine/summer2005/features/dbaker/.

Belle, J.-P. V. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. The 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries: 210-215.

Brown, C. V. (2003). "Performance Metrics for IT Human Resource Alignment." Information Systems Management 20(4): 36-42.

Brussells, S. E. (2006). "Assessment of a Government Agency's Enterprise Architecture Program." Journal of Enterprise Architecture 2(1): 43-50.

BTA. (2006). "Business Enterprise Architecture (BEA) Compliance Guidance." from http://www.dod.mil/dbt/products/investment/BEA_Compliance_Guidance_060410_FINAL.pdf.

Burk, D. (2005). "Proposed EA Assessment Framework 2.0, Architecture and Infrastructure Committee (AIC)." from colab.cim3.net/file/work/caf/meetings/mgt_2005_09_15/burk_EA_Assessment_2005_09_15.ppt.

Carbone, J. (2004). "The Case for Architecture Assessment." Retrieved 7.4., 2005, from http://www.harriskern.com/index.php?m=p&authorid=567&pid=376&aid=155.

Carbone, J. (2005). "Ten Signs of Enterprise Architecture Maturity." 2005, from http://www.harriskern.com/index.php?m=p&pid=377&authorid=&aid=190.

Chan, Y. E., S. L. Huff, D. W. Barclay and D. G. Copeland (1997). "Business Strategic Orientation, Information Systems Strategic Orientation, and Strategic Alignment." Information Systems Research 8(2): 125-150.

CIO Council (2000). Architecture Alignment and Assessment Guide, Chief Information Officers Council.

Couretas, J. M. D. K. (2005). Enterprise architectures: documentation views, evaluation tools and design alternative enumeration. IEEE International Conference on Systems, Man and Cybernetics, 2005.

Cumps, B., V. Stijn, G. Dedene and J. Vandenbulcke (2006a). An Empirical Study on Business/ICT Alignment in European Organizations. Proceedings of the 39th Hawaii International Conference on System Sciences. Kauai, Hawaii, USA, IEEE Computer Society.

Cumps, B., S. Viaene and G. Dedene (2006b). "Managing for Better Business-IT Alignment." IT Professional 8(5): 17-24.

Curran, C. (2005). "Link IT Investments to Business Metrics." Enterprise Architect 3(1): 16-18.

Dahlberg, T. and H. Kivijärvi (2006). An Integrated Framework for IT Governance and the Development and Validation of an Assessment Instrument. Proceedings of the 39th Hawaii International Conference on System Sciences. Kauai, Hawaii, USA, IEEE Computer Society.

DoC (2003). IT Architecture Capability Maturity Model. Department of Commerce (DoC).

Dunsire, K., T. O'Neill, M. Denford and J. Leaney (2005). The ABACUS architectural approach to computer-based system and enterprise evolution. 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS '05), IEEE Computer Society.

Ekstedt, M., N. Jonsson, L. Plazaola, E. S. Molina and N. Vargas (2005). An Organization-Wide Approach for Assessing Strategic Business-IT Alignment. Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET '05). Portland, USA, PICMET.

Erder, M. and P. Pureur (2003). "QFD in the Architecture Development Process." IT Professional 5(6): 44-52.

ETS Office (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

Eurocontrol. (2006). "WP 8.1.1 – Define Methodology For Validation Within OATA. Architecture Compliance Assessment Process. Edition 2.0." 2.0. from http://www.eurocontrol.int/valug/gallery/content/public/OATA-P2-D8.1.1-01%20DMVO%20Architecture%20Compliance%20Assessment%20Process.doc.

Fraser, P., J. Moultrie and M. Gregory (2002). The use of maturity models/grids as a tool in assessing product development capability. IEEE International Engineering Management Conference. Cambridge, IEEE.

GAO (2002). Enterprise Architecture Use across the Federal Government Can Be Improved. USA: Washington, D.C., United States General Accounting Office, USA.

GAO. (2003a). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." from http://www.gao.gov/new.items/d03584g.pdf.

GAO (2003b). Leadership Remains Key to Agencies Making Progress on Enterprise Architecture Efforts, General Accounting Office (GAO).

Gartner (2002). Return on Enterprise Architecture: Measure It in Asset Productivity. GartnerG2 Report. Stamford, USA, Gartner, Inc.

Government Accountability Office (GAO). (2003). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." 2004.

Hazra, T. K. (2005). "Getting Your EA Metrics Right (a presentation)." 2005.

Henderson, J. and N. Venkatraman (1990). Strategic Alignment: A model For Organizational Transformation Via Information Technology. Working Paper 3223-90. Cambridge, USA, Sloan School of Management, Massachusetts Institute of Technology.

Henderson, J. and N. Venkatraman (1992). "Aligning Business and Information Technology Domains: Strategic Planning in Hospitals." Hospital and Health Services Administrative 37(1): 71-87.

Hirvonen, A. (2005). Enterprise Architecture Planning in Practice: The Perspectives of Information and Communication Technology Service Provider and End-user. Department of Computer Science and Information Systems. Jyväskylä, University of Jyväskylä.

Hirvonen, A. and M. Pulkkinen (2003). Evaluation of IT Architecture Solutions - How Can an ICT Consultant Tell What Is Best for You? Proceeding of the 10th European Conference on Information Technology Evaluation, Management Centre International Limited.

Hite, R. (2003). Agency EA Maturity: Are We Making Progress? GAO.

Hämäläinen, N. and T. Ylimäki (2004). Architecture Management in Three IT Companies - Problems and Characteristics. Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering. EASE 2004. Edinburgh: 97-109.

Infosys. (2005). "Infosys Enterprise Architecture Survey 2005 Executive Summary." Retrieved 25 August, 2006, from http://www.infosys.com/services/systemintegration/ea-survey/ea-survey-executive-summary.pdf.

Kamogawa, T. and H. Okada (2005). A Framework for Enterprise Architecture Effectiveness. Proceedings of the Second International Conference on Services Systems and Services Management (ICSSSM '05). Chongqing, China, IEEE Computer Society.

Kluge, C., A. Dietzsch and M. Rosemann (2006). How to Realize Corporate Value from Enterprise Architecture. the Proceedings of the 14th European Conference on Information Systems (ECIS 2006). Göteborg, Sweden, Association for Information Systems.

Lee, S. M., D. L. Olson, S. Trimi and K. M. Rosacker (2005). "An Integrated Method to Evaluate Business Process Alternatives." Business Process Management Journal 11(2): 198-212.

Luftman, J. (2000). "Assessing Business-IT Alignment Maturity." Communications of the Association for Information Systems 4(Article 14).

Luftman, J. (2003). "Assessing IT/Business Alignment." Information Systems Management 20(4): 9-15.

Luftman, J. N., R. Papp and T. Brier (1999). "Enablers and Inhibitors of Business-IT Alignment." Communications of the Association for Information Systems 1(11).

McGovern, J., S. W. Ambler, M. Stevens, J. Linn, V. Sharan and E. K. Jo (2003). A Practical Guide to Enterprise Architecture. New Jersey, Prentice Hall PTR.

META Group Inc. (2000a). "Architecture Capability Assessment." META Practice 4(7).

META Group Inc. (2000b). "Architecture Maturity Assessment." METAView Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0378/mv0378.html.

META Group Inc. (2002). "Enterprise Architecture Maturity." METAView, Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0599/mv0599.html.

META Group Inc. (2004a). "Architecture Program Maturity Assessment: Findings and Trends." Enterprise Planning & Architecture Strategies, Teleconference 2118, 23 September 2004, from http://www.metagroup.com/us/displayArticle.do?oid=49449.

META Group Inc. (2004b). "Planning the Enterprise Architecture Measurement Program." META Group Research, Enterprise Planning & Architecture Strategies, Practice Summary, from http://www.metagroup.com/us/displayArticle.do?oid=50037.

Morganwalp, J. and A. P. Sage (2003). "A System of Systems Focused Enterprise Architecture Framework and an Associated Architecture Development Process." Information Knowledge Systems Management 3(2): 87-105.

Morganwalp, J. M. and A. P. Sage (2004). "Enterprise Architecture Measures of Effectiveness." International Journal of Technology, Policy and Management 4(1): 81-94.

NASCIO (2003). NASCIO Enterprise Architecture Maturity Model, v. 1.3, National Association of State Chief Information Officers (NASCIO).

NASCIO. (2004). "NASCIO Enterprise Architecture Maturity Model, v. 3.1 Self-Assessment (Draft)." 2004, from https://www.nascio.org/publications/index.cfm.

NASCIO (2005). The States and Enterprise Architecture: How far have we come? Findings from the NASCIO 2005 EA Assessment. Lexington, National Association of State Chief Information Officers (NASCIO), USA.

Niemi, E. (2006a). Architectural Work Status: Challenges and Developmental Potential - A Case Study of Three Finnish Business Enterprises. Proceedings of the 6th WSEAS International Conference on Applied Computer Science (ACS'06). Puerto de la Cruz, Tenerife, Spain, World Scientific and Engineering Society and Academy (WSEAS) Press.

Niemi, E. (2006b). Enterprise Architecture Benefits: Perceptions from Literature and Practice. Proceedings of the 7th IBIMA Conference on Internet & Information Systems in the Digital Age. Prescia, Italy, International Business Information Management Association (IBIMA).

Niemi, E. (2006c). "Enterprise Architecture Work Overview in Three Finnish Business Enterprises." WSEAS Transactions on Business and Economics 3(9): 628-635.

OMB. (2004). "Guidelines for Enterprise Architecture Assessment Framework."

OMB (2005a). Federal Enterprise Architecture Program EA Assessment Framework 2.0, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB (2005b). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB FEA Program Management Office. (2004). "Guidelines for Enterprise Architecture Assessment Framework."

OMB FEA Program Management Office (2005). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

Paras, G. (2005). Enterprise architecture: Seeing the big picture. Federal Times. Springfield, USA.

Pereira, C. M. and P. Sousa (2003). Getting into the misalignment between Business and Information Systems. Proceedings of the 10th European Conference on Information Technology Evaluation. Madrid, Spain, Academic Conferences International.

Rehkopf, T. W. and N. Wybolt (2003). "Top 10 Architecture Land Mines." IT Professional 5(6): 36-43.

Rico, D. F. (2005). "A Framework for Measuring the ROI of Enterprise Architecture."

Rosenfeld, L., K. Instone, S. Cupala, J. J. Garrett, M. Hearst, G. Marchionini and N. Ragouzis (2001). Measuring Information Architecture Quality: Prove It (or not)! CHI '01 extended abstracts on Human factors in computing systems. Seattle, Washington, USA, ACM Press: 219-220.

Rosser, B. (2006). Measuring the Value of Enterprise Architecture: Metrics and ROI, Gartner.

Saha, P. (2006). "A Real Options Perspective to Enterprise Architecture as an Investment Activity." Journal of Enterprise Architecture 2(3): 32-52.

Saleh, Y. and M. Alshawi (2005). "An alternative model for measuring the success of IS projects: the GPIS model." Journal of Enterprise Information Management 18(1): 47-63.

Schekkerman, J. (2003). "Extended Enterprise Maturity Model (E2AMM)." from http://www.enterprise-architecture.info/Images/E2AF/E2AMMv2.PDF.

Schekkerman, J. (2004a). Enterprise Architecture Validation - Achieving Business-Aligned and Validated Enterprise Architectures, Institute For Enterprise Architecture Developments, URL: http://www.enterprise-architecture.info/: 27.

Schekkerman, J. (2004b). How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework. Canada, Trafford.

Schekkerman, J. (2004a). Enterprise Architecture Validation - Achieving Business-Aligned and Validated Enterprise Architectures, Institute For Enterprise Architecture Developments, URL: http://www.enterprise-architecture.info/: 27.

Schekkerman, J. (2004b). How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework. Canada, Trafford.

Schekkerman, J. (2005a). The Economic Benefits of Enterprise Architecture. New Bern, USA, Trafford.

Schekkerman, J. (2005b). "Trends in Enterprise Architecture 2005 - How are Organizations Progressing? Web-form Based Survey 2005." Retrieved August 15, 2006, from http://www.enterprise-architecture.info/Images/EA%20Survey/Enterprise%20Architecture%20Survey%202005%20IFEAD%20v10.pdf.

Schmidt, J. (2005). "Valuing Enterprise Architecture." Online Features Retrieved 21.8., 2006, from http://www2.darwinmag.com/read/feature/jan05_eavalue.cfm.

Simonsson, M., Å. Lindström, P. Johnson, L. Nordström, J. Grundbäck and O. Wijnbladh (2005). Scenario-Based Evaluation of Enterprise Architecture. A Top-Down Approach for Chief Information Officer Decision Making. Proceedings of the International Conference on Enterprise Information Systems (ICEIS). Miami, USA.

Syntel. (2005). "Evaluating Your Enterprise Architecture." Applications White Paper Series Retrieved 22 August, 2006, from http://www.syntelinc.com/uploadedfiles/Syntel_EvaluateEnterArchit.pdf.

Tang, A., J. Han and P. Chen (2004). A Comparative Analysis of Architecture Frameworks. Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04). Busan, Korea, IEEE Computer Society.

Teo, T. S. H. and J. S. K. Ang (1999). "Critical success factors in the alignment of IS plans with business plans." International Journal of Information Management 19(2): 173-185.

The Office of Enterprise Technology Strategies (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

The Open Group. (2006). "The Open Group Architecture Framework version 8.1.1, Enterprise Edition (TOGAF 8.1.1)." Retrieved 10 September, 2006, from http://www.opengroup.org/architecture/togaf/.

Van Belle, J. P. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. Proceedings of the 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (SAICSIT-2004): 210-215.

van der Raadt, B., J. Hoorn, F. and H. van Vliet (2005). Alignment and Maturity are Siblings in Architecture Assessment. Proceedings of the 17th Conference on Advanced Information Systems Engineering, CAiSE 2005. Porto, Portugal, Springer.

van der Raadt, B., J. Soetendal, M. Perdeck and H. v. Vliet (2004). Polyphony in Architecture. Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), IEEE Computer Society: 533-542.

Whitman, L., K. Ramachandran and V. Ketkar (2001). A Taxonomy of a Living Model of the Enterprise. Proceedings of the 2001 Winter Simulation Conference. B. A. Peters, Smith, J.S., Medeiros, D.J., Rohrer, M.W. Arlington, Virginia, IEEE Computer Society: 848-855.

Ylimäki, E. N. a. T. (2007). Enterprise Architecture Evaluation Components. Abstract accepted to HAAMAHA 2007.

Ylimäki, T. (2006a). "Potential Critical Success Factors for Enterprise Architecture." Journal of Enterprise Architecture 2(4): 29-40.

Ylimäki, T. (2006b). Towards a Generic Evaluation Model for Enterprise Architecture. Submitted to the Journal of Enterprise Architecture. Information Technology Research Institute, University of Jyväskylä, Finland.

Ylimäki, T. and E. Niemi (2006). Evaluation Needs for Enterprise Architecture. AISA Project Report. Jyväskylä, Finland, Information Technology Research Institute, University of Jyväskylä.

1.3 Software Architecture Evaluation

As with enterprise architecture evaluation, this category includes general software architecture evaluation literature. In this category, the focus is also on evaluation approaches and methods, with some discussion on metrics. In the software architecture domain, the evaluation research is more established than in the enterprise architecture domain, with some extensively used approaches such as ATAM. In this domain, the concept of maturity is substituted with more accurate measures.

Abowd, G., L. Bass, P. Clements, R. Kazman, L. Northrop and A. Zaremski (1997). Recommended Best Industrial Practice for Software Architecture Evaluation, Technical Report CMU/SEI-96-TR-025, Software Engineering Institute (SEI), Carnegie Mellon University.

Abowd, G. D. (1995). Defining reference models and software architectural styles for cooperative systems. ACM SIGOIS Bulletin, Special issue: workshop write-ups and positions papers from CSCW'94.

Al-Naeem, T., I. Gorton, F. Rabhi and B. Benatallah (2005). Tool support for optimization-based architectural evaluation. Proceedings of the second international workshop on Models and processes for the evaluation of off-the-shelf components. St. Louis, Missouri, ACM Press.

AlSharif, M., W. P. Bond and T. Al-Otaiby (2004). Assessing the complexity of software architecture. Proceedings of the 42nd annual Southeast regional conference. Huntsville, Alabama, USA, ACM Press: 98-103.

Asundi, J., R. Kazman and M. Klein (2001). Using Economic Considerations to Choose Among Architecture Design Alternatives, The Software Engineering Institute, Carnegie Mellon University.

Avritzer, A. and E. J. Weyuker (1998). Investigating Metrics for Architectural Assessment. Proceedings of the Fifth International Software Metrics Symposium, Metrics 1998. Bethesda, MD, IEEE Computer Society: 4-10.

Avritzer, A. and E. J. Weyuker (1999). "Metrics to Assess the Likelihood of Project Success Based on Architecture Reviews." Empirical Software Engineering 4(3): 199 - 215.

Babar, M. A. and I. Gorton (2004). Comparison of Scenario-Based Software Architecture Evaluation Methods. Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSE'04), IEEE Computer Society: 600-607.

Babar, M. A., L. Zhu and R. Jeffery (2004). A Framework for Classifying and Comparing Software Architecture Evaluation Methods. Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), IEEE Computer Society.

Bachmann, F., L. Bass and M. Klein (2002). Illuminating the Fundamental Contributors to Software Architecture Quality, The Software Engineering Institute, Carnegie Mellon University.

Bahsoon, R. and W. Emmerich (2003). Evaluating Software Architectures: Development, Stability, and Evolution. Proceedings of ACS/IEEE International Conference on Computer Systems and Applications. Tunis, Tunisia: 47-57.

Ballantine, J., M. Bonner and M. Levy (1998). Developing a 3-D Model of Information Systems Success. Information Systems Success Measurement. E. J. Garrity and G. L. Sanders. Hershey, USA, Idea Group Publishing.

Balsamo, S., P. Inverardi and C. Mangano (1998). An approach to performance evaluation of software architectures. Proceedings of the 1st international workshop on Software and performance WOSP '98, Santa Fe, New Mexico, United States, ACM Press.

Barbacci, M., P. Clements, A. Lattanze, L. Northrop and W. Wood (2003a). Using the Architecture Tradeoff Analysis Method (ATAM) to Evaluate the Software Architecture for a Product Line of Avionics Systems: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M., R. Ellison, C. Weinstock and W. Wood (2000). Quality Attribute Workshop Participant's Handbook, Technical Report CMU/SEI-2000-SR-001, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R. (2002). SEI Architecture Analysis Techniques and When to Use Them, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., S. J. Carriere, P. H. Feiler, R. Kazman, M. H. Klein, H. F. Lipson, T. A. Longstaff and C. B. Weinstock (1997a). Steps in an Architecture Tradeoff Analysis Method: Quality Attribute Models and Analysis, Technical Report CMU/SEI-97-TR-029, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., R. Ellison, A. J. Lattanze, J. A. Stafford, C. B. Weinstock and W. G. Wood (2003b). Quality Attribute Workshops (QAWs), Third Edition, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., R. Ellison, J. A. Stafford, C. B. Weinstock and W. G. Wood (2001). Quality Attribute Workshops, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., M. H. Klein and C. B. Weinstock (1997b). Principles for Evaluating the Quality Attributes of a Software Architecture, Technical Report CMU/SEI-96-TR-036, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R. and W. G. Wood (1999). Architecture Tradeoff Analyses of C4ISR Products, The Software Engineering Institute, Carnegie Mellon University.

Barber, K. S. and J. Holt (2001). "Software Architecture Correctness." IEEE Software 18(6): 64-65.

Barber, K. S., J. Holt and G. Baker (2002). Performance evaluation of domain reference architectures. Proceedings of the 14th international conference on Software engineering and knowledge engineering SEKE '02, Ischia, Italy, ACM Press.

Barber, K. S. G., T.J. (2000). Tool support for systematic class identification in object-oriented software architectures. Proceedings. 37th International Conference on Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Sydney, NSW.

Bass, L. and B. E. John (2000). Achieving usability through software architectural styles. CHI '00 extended abstracts on Human factors in computing systems, The Hague, The Netherlands, ACM Press.

Bass, L. and B. E. John (2001). "Evaluating Software Architectures for Usability." Lecture Notes in Computer Science 2254.

Bass, L., B. E. John and J. Kates (2001a). Achieving Usability Through Software Architecture, The Software Engineering Institute, Carnegie Mellon University.

Bass, L. and R. Kazman (1999). Architecture-Based Development, The Software Engineering Institute, Carnegie Mellon University.

Bass, L., M. Klein and G. Moreno (2001b). Applicability of General Scenarios to the Architecture Tradeoff Analysis Method, The Software Engineering Institute, Carnegie Mellon University.

Benarif, S., A. Ramdane-Cherif, N. Levy and F. Losavio (2004). Intelligent Tool Based-Agent for Software Architecture Evaluation. Proceedings of the Fourth International Conference on Quality Software (QSIC'04), IEEE Computer Society: 126-133.

Bengtsson, P. (1999). Design and Evaluation of Software Architecture. Department of Software Engineering and Computer Science. Karlskrona, University of Karlskrona/Ronneby.

Bengtsson, P. and J. Bosch (1998). Scenario-Based Software Architecture Reengineering. Proceedings of the 5th International Conference on Software Reuse. Victoria, BC, IEEE Computer Society: 308-317.

Bengtsson, P. and J. Bosch (1999). Architecture Level Prediction of Software Maintenance. Proceedings of the Third European Conference on Software Maintenance and Reengineering. Amsterdam, IEEE Computer Society: 139-147.

Bengtsson, P., N. Lassing, J. Bosch and H. van Vliet (2004). "Architecture-Level Modifiability Analysis (ALMA)." Journal of Systems and Software 69(1-2): 129-147.

Bergey, J., M. Barbacci and W. Wood (2000). Using Quality Attribute Workshops to Evaluate Architectural Design Approaches in a Major System Acquisition: A Case Study, Technical Note CMU/SEI-2000-TN-010, Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and P. C. Clements (2005). Software Architecture in DoD Acquisition: A Reference Standard for a Software Architecture Document, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and M. J. Fisher (2001). Use of the Architecture Tradeoff Analysis Method (ATAM) in the Acquisition of Software-Intensive Systems, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K., M. J. Fisher and L. G. Jones (2002). Use of the Architecture Tradeoff Analysis Method (ATAM) in Source Selection of Software-Intensive Systems, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K., M. J. Fisher, L. G. Jones and R. Kazman (1999). Software Architecture Evaluation with ATAM in the DoD System Acquisition Context, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and W. G. Wood (2002). Use of Quality Attribute Workshops (QAWs) in Source Selection for a DoD System Acquisition: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Bleistein, S. J., K. Cox and J. Verner (2006). "Validating strategic alignment of organizational IT requirements using goal modeling and problem diagrams." Journal of Systems and SOftware 79(3): 362-378.

Boehm, B. (1994). Software architectures: critical success factors and cost drivers. Proceedings of the 16th International Conference on Software Engineering. Sorrento, Italy, IEEE Computer Society: 365.

Boehm, B. (1996). "Anchoring the software process." IEEE Software 13(4): 73-82.

Bokhari, R. H. (2005). "The relationship between system usage and user satisfaction: a meta-analysis." The Journal of Enterprise Information Management.

Bosch, J. and P. Molin (1999). Software Architecture Design: Evaluation and Transformation. Proceedings of the IEEE Conference and Workshop on Engineering of Computer-Based Systems, ECBS '99. Nashville, TN, USA, IEEE Computer Society: 4-10.

Bot, S., C.-H. Lung and M. Farrell (1996). A Stakeholder-Centric Software Architecture Analysis Approach. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, USA, ACM Press: 152-154.

Bredemeyer Consulting. (2000). "Software Architecting Success Factors and Pitfalls." from http://www.bredemeyer.com/CSFs_pitfalls.htm.

Briand, L. C., J. Carrière, R. Kazman and J. Wüst (1998a). A Comprehensive Framework for Architecture Evaluation. Technical Report ISERN-98-28. Kaiserslautern, Germany, Fraunhofer Institute For Experimental Software Engineering.

Briand, L. C., S. J. Carrière, R. Kazman and J. Wüst (1998b). COMPARE: A Comprehensive Framework for Architecture Evaluation. ECOOP'98 Workshop Reader, Lecture Notes in Computer Science (LNCS). S. Demeyer and J. Bosch, Springer-Verlag. 1543: 48-49.

Choi, H. and K. Yeom (2002). An Approach to Software Architecture Evaluation with the 4+1 View Model of Architecture. Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02), IEEE Computer Society: 286-293.

Chrissis, M. B., M. Konrad and S. Shrum (2003). CMMI: Guidelines for Process Integration and Product Improvement, Addison Wesley.

Chung, L., B. A. Nixon and E. Yu (1995). An Approach to Building Quality into Software Architecture. Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research. Toronto, Ontario, Canada, IBM Press.

Clements, P., R. Kazman and M. Klein (2001). Evaluating Software Architectures: Methods and Case Studies. Boston, USA, Addison-Wesley.

Clements, P. C. (2000). Active Reviews for Intermediate Designs, CMU/SEI-2000-TN-009, Software Engineering Institute (SEI), Carnegie Mellon University.

DeLone, W. and E. McLean (1992). "Information Systems Success: The Quest for the Dependent Variable." Information Systems Reseach 3(1): 60-95.

DeLone, W. and E. McLean (2003). "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update." Journal of Management Information Systems 19(4): 9-30.

DeLone, W. H. and E. R. McLean (2002). Information Systems Success Revisited. Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02). Big Island, Hawaii, USA, IEEE Computer Society.

DeLone, W. H. and E. R. McLean (2004). "Measuring e-Commerce Success: Applying the DeLOne & McLean Information Systems Success Model." International Journal of Electronic Commerce 9(1): 31-47.

Dias, M. S. and M. E. R. Vieira (2000). Software Architecture Analysis Based on Statechart Semantics. Proceedings of the Tenth International Workshop on Software Specification and Design. San Diego, CA, IEEE Computer Society: 133-137.

Dias, O. P., I. C. Teixeira and J. P. Teixeira (1999). "Metrics and Criteria for Quality Assessment of Testable Hw/Sw Systems Architectures." Journal of Electronic Testing: Theory and Applications 14: 149-158.

Dikel, D., D. Kane, S. Ornburn and J. Wilson. (1995). "Software Architecture Case Study: Organizational Success Factors." from http://www.vraps.com/files/archcase_exec_sum.doc http://www.vraps.com/files/.

Dobrica, L. and E. Niemelä (2002). "A Survey on Software Architecture Analysis Methods." IEEE Transactions on Software Engineering 28(7): 638-653.

Dolan, T., J. (2001). Architecture Assessment of Information-System Families. Eindhoven, The Netherlands, Technische Universiteit Eindhoven: 222.

Dolan, T., R. Weterings and J. C. Wortmann (2000). Stakeholder-Centric Assessment of Product Family Architecture. Practical Guidelines for Information System Interoperability and Extensibility. Proceedings of the International Workshop on Software Architectures for Product Families (IW-SAPF-3), Lecture Notes in Computer Science (LNCS). F. v. d. Linden. Las Palmas de Gran Canaria, Spain, Springer-Verlag. 1951: 225-243.

Drudis, A. and J. Mendonca. "The Architecture of Web Applications." Retrieved 26.5.2005, from http://www.interex.org/pubcontent/enterprise/may01/14drudis.html.

Dueñas, J. C., W. L. d. Oliveira and J. A. d. l. Puente (1998). "A Software Architecture Evaluation Model." Lecture Notes in Computer Science 1429: 148-157.

Eickelmann, N. S. and D. J. Richardson (1996). An evaluation of software test environment architectures. Proceedings of the 18th international conference on Software engineering, Berlin, Germany, IEEE Computer Society.

Elpez, I. and D. Fink (2006). "Information Systems Success in the Public Sector: Stakeholder's Perspectives and Emerging Alignment Model." Issues in Informing Science and Information Technology 3(1).

Evans, C. and E. Shiu (2006). "E-business cross-functional alignment: the integration of processes and applications." International Journal of Electronic Business 4(3/4): 302-319.

Everitt, T., R. Tvedt, T. and J. Tvedt, D. (2004). Validating and Improving an Existing Software Architectural Evaluation Process. Proceedings of the 20th IEEE International Conference on Software Maintenance (ICSM'04): 417-421.

Fabbrini, F., M. Fusani and S. Gnesi (1998). Quality Evaluation based on Architecture Analysis. Proceedings of the International Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA'98). Marsala, Italy.

Folmer, E. and J. Bosch (2004). "Architecting for Usability: a Survey." The Journal of Systems and Software 70: 61-78.

Folmer, E., J. van Gurp and J. Bosch (2003). Scenario-Based Assessment of Software Architecture Usability. Proceedings of the International Conference on Software Engineering (ICSE 2003). Portland, Oregon, USA: 61-68.

Fraser, P., J. Moultrie and M. Gregory (2002). The use of maturity models/grids as a tool in assessing product development capability. IEEE International Engineering Management Conference. Cambridge, IEEE.

Fukuzawa, K. and M. Saeki (2002). Evaluating Software Architectures by Coloured Petri Nets. Proceedings of the 14th international conference on Software engineering and knowledge engineering. Ischia, Italy, ACM Press: 263-270.

Gable, G. G., D. Sedera and T. Chan (2003). Enterprise Systems Success: A Measurement Model. Proceedings of the 24th International Conference on Information Systems. Seattle, USA, Association for Information Systems.

Gallagher, B. P. (2000). Using the Architecture Tradeoff Analysis Method to Evaluate a Reference Architecture: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Goedicke, M. and T. Meyer (1998). WWW-based software architecture design support for cooperative representation and checking. Proceedings of the third international workshop on Software architecture, Orlando, Florida, United States, ACM Press.

Golden, E. J., B.E.; Bass, L. (2005). Quality vs. quantity: comparing evaluation methods in a usability-focused software architecture modification task. International Symposium on Empirical Software Engineering.

Gorton, I. and L. Zhu (2005). Tool support for just-in-time architecture reconstruction and evaluation: an experience report. Proceedings of the 27th international conference on Software engineering, St. Louis, MO, USA, ACM Press.

Grahn, H., M. Mattsson and F. Mårtensson (2003). An approach for performance evaluation of software architectures using prototyping. 7th IASTED International Conference on Software Engineering and Applications.

Grassi, V. M., R. (2003). "Derivation of Markov models for effectiveness analysis of adaptable software architectures for mobile computing." IEEE Transactions on Mobile Computing 2(2): 114 - 131.

Gustafsson, J., J. Paakki, L. Nenonen and A. I. Verkamo (2002). Architecture-Centric Software Evolution by Software Metrics and Design Patterns. Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02). Budapest, IEEE Computer Society: 108-115.

Haley, G. (2005). "Software Architecture Evaluation - Transforming a craft into a business process." Retrieved 26.5., 2005, from http://sunset.usc.edu/gsaw/gsaw2005/s9b/haley.pdf.

Hilliard, R., M. Kurland, J., S. Litvintchouk, D., T. Rice and S. Schwarm (1996). Architecture Quality Assessment, version 2.0, The MITRE Corporation.

Hilliard, R., M. J. Kurland and S. D. Litvintchouk (1997). MITRE's Architecture Quality Assessment. Proceedings of the Software Engineering & Economics Conference.

Hu, P. J.-H. (2002). Evaluating Telemedicine Systems Success: A Revised Model. Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'02). Big Island, Hawaii, USA, IEEE Computer Society.

Hämäläinen, N., J. Ahonen and T. Kärkkäinen (2005). Why to Evaluate Enterprise and Software Architectures - Objectives and Use Cases. Proceeding of the 12th European Conference on Information Technology Evaluation, Turku.

Hämäläinen, N. and T. Ylimäki (2004). Architecture Management in Three IT Companies - Problems and Characteristics. Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering. EASE 2004. Edinburgh: 97-109.

Ifinedo, P. and N. Nahar (2006). Prioritization of Enterprise Resource Planning (ERP) Systems Success Measures: Viewpoints of Two Organizational Stakeholder Groups. Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC'06). Dijon, France, ACM Press.

Ifinedo, P. E. (2006). Enterprise Resource Planning Systems Success Assessment: An Integrative Framework. Department of Computer Science and Information Systems. Jyväskylä, FInland, University of Jyväskylä. PhD: 153.

Iivari, J. (2005). "An Empirical Test of the DeLone-McLean Model of Information System Success." The DATA BASE for Advances in Information Systems 36(2): 8-27.

Ionita, M. T., P. America, D. K. Hammer, H. Obbink and J. J. M. Trienekens (2004). A scenario-driven approach for value, risk, and cost analysis in system architecting for innovation. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Ionita, M. T., D. K. Hammer and H. Obbink (2002). Scenario-Based Software Architecture Evaluation Methods: An Overview. Proceedings of the International Conference on Software Engineering (ICSE 2002). Orlando, Florida, USA.

Jones, L. G. and A. J. Lattanze (2001). Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R. (1996). Tool Support for Architecture Analysis and Design. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, California, United States: 94-97.

Kazman, R., G. Abowd, L. Bass and P. Clements (1996). "Scenario-Based Analysis of Software Architecture." IEEE Software 13(6): 47-55.

Kazman, R., J. Asundi and M. Klein (2001). Quantifying the Costs and Benefits of Architectural Decisions. Proceedings of the 23rd International Conference on Software Engineering. Toronto, Ontario, Canada, IEE Computer Society: 297-306.

Kazman, R., M. Barbacci, M. Klein, S. J. Carrière and S. G. Woods (1999). Experience with Performing Architecture Tradeoff Analysis. Proceedings of the 21st international conference on Software engineering. Los Angeles, California, United States, IEEE Computer Society Press: 54-63.

Kazman, R. and L. Bass (2002). "Making Architecture Reviews Work in the Real World." IEEE Software 19(1): 67-73.

Kazman, R., L. Bass, G. Abowd and M. Webb (1994). SAAM: A Method for Analyzing the Properties of Software Architectures. Proceedings of the 16th International Conference on Software Engineering: 81-90.

Kazman, R., L. Bass, G. Abowd and M. Webb (1998a). SAAM: A Method for Analyzing the Properties of Software Architectures. Proceedings of the 16th International Conference on Software Engineering: 81-90.

Kazman, R. and S. J. Carriere (1997). Playing Detective: Reconstructing Software Architecture from Available Evidence, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R. and M. Klein (1998). Performing Architecture Tradeoff Analysis. Proceedings of the third international workshop on software architecture. Orlando, Florida, USA, ACM Press: 85-88.

Kazman, R., M. Klein, M. Barbacci, T. Longstaff, H. Lipson and J. Carriere (1998b). The Architecture Tradeoff Analysis Method, Technical Report CMU/SEI-98-TR-008, Software Engineering Institute, Carnegie Mellon University.

Kazman, R., M. Klein and P. Clements (2000). ATAM: Method for Architecture Evaluation, Technical Report CMU/SEI-2000-TR-004, Software Engineering Institute, Carnegie Mellon University.

Kazman, R., P. Kruchten, R. L. Nord and J. E. Tomayko (2004). Integrating Software-Architecture-Centric Methods into the Rational Unified Process, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R., R. L. Nord and M. Klein (2003a). A Life-Cycle View of Architecture Analysis and Design Methods, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R., L. O'Brien and C. Verhoef (2003b). Architecture Reconstruction Guidelines, Third Edition, The Software Engineering Institute, Carnegie Mellon University.

Khajenoori, S., L. Prem, K. Stevens, B. S. Keng and N. Kameli (2004). "Knowledge Centered Assessment Pattern: an Effective Tool for Assessing Safety Concerns in Software Architecture." The Journal of Systems and Software 73: 313-322.

Khalifa, M. and V. Liu (2004). "The State of Research on Information System Satisfaction." Journal of Information Technology Theory and Application 5(4): 37-50.

Knodel, J. L., M.; Muthig, D. (2005). Static Evaluation of Software Architectures - A Short Summary. 5th Working IEEE/IFIP Conference on Software Architecture, WICSA 2005.

Kornecki, A. J. and J. Zalewski. (2004). "TUTORIAL: Software Development for Real-Time Safety-Critical Applications." from http://ieeexplore.ieee.org/iel5/9714/30657/01416838.pdf?arnumber=1416838.

Krueger, I. H. (2004). Evaluating Software Architectures: Stakeholders, Metrics, Results, Migration Strategies. Department of Computer Science & Engineering. San Diego, University of California.

Lassing, N., P. Bengtsson, H. van Vliet and J. Bosch (2002a). "Experiences with ALMA: Architecture-Level Modifiability Analysis." Journal of Systems and Software 61(1): 47-57.

Lassing, N., D. Rijsenbrij and H. van Vliet (1999a). The Goal of Software Architecture Analysis: Confidence Building or Risk Assessment. Proceedings of the First Benelux Conference on State-of-the-Art of ICT Architecture.

Lassing, N., D. Rijsenbrij and H. van Vliet (1999b). On Software Architecture Analysis of Flexibility; Complexity of Changes: Size isn't Everything. Proceedings of the Second Nordic Workshop on Software Architecture (NOSA99).

Lassing, N., D. Rijsenbrij and H. van Vliet (1999c). Towards a Broader View on Software Architecture Analysis of Flexibility. Proceedings of the Sixth Asia Pacific Software Engineering Conference (APSEC '99). Takamatsu, IEEE Computer Society: 238-245.

Lassing, N., D. Rijsenbrij and H. van Vliet (2002b). Experience with ALMA. Proceedings ICSE 2002 Workshop on Methods and Techniques for Software Architecture Review and Assessment, IEEE Computer Society: 15-18.

Lehto, J., A. and P. Marttiin (2005). Experiences in System Architecture Evaluation: A Communication View for Architectural Design. Proceedings of the 38th Hawaii

International Conference on System Sciences (HICSS'05), IEEE Computer Society: 312c-312c.

Lindvall, M., R. Tesoriero and P. Costa (2002). Avoiding Architectural Degeneration: An Evaluation Process for Software Architecture. Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS'02), IEEE Computer Society: 77-86.

Lindvall, M., R. T. Tvedt and P. Costa (2003). "An Empirically-Based Process for Software Architecture Evaluation." Empirical Software Engineering 8(1): 83-108.

Lopez, M. (2000). An Evaluation Theory Perspective of the Architecture Tradeoff Analysis Method (ATAM). Pittsburg, USA, The Software Engineering Institute, Carnegie Mellon University.

Losavio, F., L. Chirinos, N. Lévy and A. Ramdane-Cherif (2003). "Quality Characteristics for Software Architecture." Journal of Object Technology 2(2): 133-150.

Losavio, F., L. Chirinos, A. Matteo, N. Lévy and A. Ramdane-Cherif (2004). "ISO Quality Standards for Measuring Architectures." The Journal of Systems and Software 72: 209-223.

Losavio, F., A. Matteo, J. O. Ordaz, N. Levy and R. Marcano-Kamenoff (2002). Quality Characteristics to Select an Architecture for Real-time Internet Applications. 4th International Software Quality Week Europe (QWE2000). Brussels, Belgium.

Lung, C.-H., S. Bot, K. Kalaichelvan and R. Kazman (1997). An Approach to Software Architecture Analysis for Evolution and Reusability. Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research. Toronto, Ontario, Canada, IBM Press: 15-26.

Lung, C.-H. and K. Kalaichelvan (2000). "An Approach to Quantitative Software Architecture Sensitivity Analysis." International Journal of Software Engineering and Knowledge Engineering 10(1): 97-114.

Maccari, A. (2002). Experiences in Assessing Product Family Software Architecture for Evolution. Proceedings of the 24th International Conference on Software Engineering. Orlando, Florida, USA, ACM Press: 585-592.

Maranzano, J. F., S. A. Rozsypal, G. H. Zimmerman, G. W. Warnken, P. E. Wirth and D. M. Weiss (2005). "Architecture Reviews: Practice and Experience." IEEE Software 22(2): 34-43.

Matinlassi, M. (2004a). Comparison of Software Product Line Architecture Desing Methods: COPA, FAST, FORM, KobrA and QADA. Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE Computer Society.

Matinlassi, M. (2004b). Evaluating the portability and maintainability of software product family architecture: terminal software case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Matinlassi, M., E. Niemelä and L. Dobrica (2002). Quality-Driven Architecture Design and Quality Analysis Method - A Revolutionary Initiation Approach to a Product Line Architecture. VTT Publications. Espoo, Technical Research Centre of Finland, VTT: 128.

McCay, B. M. (1996). Some Thoughts on the Quality of a Computer-Based System's Architecture. Engineering of the IEEE Symposium and Workshop on Computer-Based Systems. Friedrichshafen, IEEE Computer Society: 228-234.

McGovern, J., S. W. Ambler, M. Stevens, J. Linn, V. Sharan and E. K. Jo (2003). A Practical Guide to Enterprise Architecture. New Jersey, Prentice Hall PTR.

Melone, N. P. (1990). "A Theoretical Assessment of the User-Satisfaction Construct in Information Systems Research." Management Science 36(1): 76-91.

Molla, A. and P. S. Licker (2001). "E-Commerce Systems Success: An Attempt to Extend and Respecify the DeLone and MacLean Model of IS Success." Journal of Electronic Commerce Research 2(4): 131-141.

Moore, M., R. Kazman, M. Klein and J. Asundi (2003). Quantifying the Value of Architecture Design Decisions: Lessons from the Field. Proceedings of the 25th International Conference on Software Engineering, IEEE Computer Society: 557-562.

Morisio, M., I. Stamelos and A. Tsoukias (2002). A new method to evaluate software artifacts against predefined profiles. Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE 2002). Ischia, Italy, ACM Press: 811-818.

Niemelä, E., M. Matinlassi and A. Taulavuori (2004). "Practical Evaluation of Software Product Family Architectures." Lecture Notes in Computer Science 3154: 130-145.

Nord, R. L., M. R. Barbacci, P. Clements, R. Kazman, M. Klein, L. O'Brien and J. E. Tomayko (2003). Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM), The Software Engineering Institute, Carnegie Mellon University.

Paulk, M. C., B. Curtis, M. B. Chrissis and C. V. Weber (1993). Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-024-ESC-TR-93-177, SEI.

Petriu, D. S., C.; Jalnapurkar, A. (2000). "Architecture-based performance analysis applied to a telecommunication system." IEEE Transactions on Software Engineering 26(11): 1049 - 1065.

Purhonen, A. (2004). Performance optimization of embedded software architecture - a case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Rai, A., S. S. Lang and R. B. Welker (2002). "Assessing the Validity of IS Success Models: An Empirical Test and Theoretical Analysis." Information Systems Research 13(1): 50-69.

Ran, A. (2001). Tutorial on fundamental concepts for practical software architecture. Proceedings of the 23rd International Conference on Software Engineering, Toronto, Ontario, Canada, IEEE Computer Society.

Ran, A. (2001). Fundamental concepts for practical software architecture. Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT

international symposium on Foundations of software engineering ESEC/FSE-9, Vienna, Austria, ACM Press.

Richardson, D. J. and A. L. Wolf (1996). Software Testing at the Architectural Level. Joint Proceedings of the Second International Software Architecture Workshop (ISAW-2) and International Workshop on Multiple Perspectives in Software Development (Viewpoints '96) on SIGSOFT '96 Workshops. San Francisco, California, United States, ACM Press: 68-71.

Scallon, G. (1999). "Model Based Development." Retrieved 26.5.2005, from http://www.stc-online.org/cd-rom/1999/slides/MBDslid1.pdf.

Seddon, P. B. (1997). "A Respecification and Extension of the DeLone and McLean Model of IS Success." Information Systems Research 8(3): 240-253.

Seddon, P. B., S. Staples, R. Patnayakuni and M. Bowtell (1999). "Dimensions of Information Systems Success." Communications of the Association for Information Systems 2(20).

Sharma, V. S. and K. S. Trivedi (2005). Architecture based analysis of performance, reliability and security of software systems. Proceedings of the 5th international workshop on Software and performance WOSP '05, Palma, Illes Balears, Spain, ACM Press.

Shereshevsky, M., H. Ammari, N. Gradetsky, A. Mili and H. H. Ammar (2001). Information Theoretic Metrics for Software Architectures. Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC 2001), Chicago, IL, USA, IEEE Computer Society.

Simone, M. D. and R. Kazman (1995). Software architectural analysis: an experience report. Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, IBM Press.

Stoermer, C., F. Bachmann and C. Verhoef (2003). SACAM: The Software Architecture Comparison Analysis Method, The Software Engineering Institute, Carnegie Mellon University.

Svahnberg, M. and C. Wohlin (2002). Consensus Building when Comparing Software Architectures. Proceedings of the 14th International Conference on Product Focused Software Process Improvement (PROFES 2002), Lecture Notes in Computer Science (LNCS). M. Oivo and S. Komi-Sirviö. Rovaniemi, Finland, Springer. 2559: 436-452.

Svahnberg, M. and C. Wohlin (2005). "An Investigation of a Method for Identifying a Software Architecture Candidate with Respect to Quality Attributes." Empirical Software Engineering 10: 149-181.

Svahnberg, M., C. Wohlin, L. Lundberg and M. Mattsson (2002). A Method for Understanding Quality Attributes in Software Architecture Structures. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering. Ischia, Italy, ACM Press: 819-826.

Tahvildari, L., K. Kontogiannis and J. Mylopoulos (2003). "Quality-Driven Software Re-engineering." The Journal of Systems and Software 66: 225-239.

Taylor, R. N. (1996). Generalization from domain experience: the superior paradigm for software architecture research? Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops, San Francisco, California, United States, ACM Press.

Tekinerdogan, B. (2004). ASAAM: aspectual software architecture analysis method. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Tvedt, R. T., M. Lindvall and P. Costa (2002). A Process for Software Architecture Evaluation Using Metrics. Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02): 191-196.

van der Raadt, B., J. Soetendal, M. Perdeck and H. v. Vliet (2004). Polyphony in Architecture. Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), IEEE Computer Society: 533-542.

van Gurp, J. B., J. (2000). SAABNet: Managing qualitative knowledge in software architecture assessment. Proceedings. Seventh IEEE International Conference and Workshopon the Engineering of Computer Based Systems, 2000. (ECBS 2000). Edinburgh.

Vigder, M., T. McClean and F. Bordeleau (2003). "Evaluating COTS Based Architectures." Lecture Notes in Computer Science 2580: 240 - 250.

Wilkin, C. and B. Hewett (1999). Quality in a Respecification of DeLone and McLean's IS Success Model. Managing Information Technology Resources in Organizations in the Next Millennium. M. Khosrowpour. Hershey, USA, Idea Group Publishing: 663-672.

Williams, L. G. and C. U. Smith (1998). Performance evaluation of software architectures. Proceedings of the 1st international workshop on Software and performance WOSP '98, Santa Fe, New Mexico, United States, ACM Press.

Williams, L. G. and C. U. Smith (2002). PASA: A Method for the Performance Assessment of Software Architectures. Proceedings of the 3rd Workshop on Software Performance, WOSP '02. Rome, Italy: 179-189.

Woods, S. G. and M. Barbacci (1999). Architectural Evaluation of Collaborative Agent-Based Systems, The Software Engineering Institute, Carnegie Mellon University.

Wu, C. C., E. (2005). Comparison of Web service architectures based on architecture quality properties. 3rd IEEE International Conference on Industrial Informatics, INDIN '05.

Zeng, D. D. and J. L. Zhao (2002). Achieving Software Flexibility via Intelligent Workflow Techniques. Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02), IEEE Computer Society: 606-615.

Zhu, L., M. A. Babar and R. Jeffery (2004). Mining Patterns to Support Software Architecture Evaluation. Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04), IEEE Computer Society: 25-34.

1.4 Enterprise Architecture Success Factors, Evaluation Criteria and Metrics

In this category, a more in-depth view is taken on enterprise architecture evaluation and various success factors, evaluation criteria and metrics introduced. Enterprise architecture success factors represent areas where work has to be carried out exceptionally well to enable high-quality architecture - areas such as top management support and communication are typically mentioned. In general, the critical success factors approach provides a high-level view on enterprise architecture and enterprise architecture work quality. Evaluation criteria and metrics, on the other hand, can be related to the quality of various architectural views, especially software or system architecture, or the quality of architecture work, such as communication or business-IT alignment. Maturity evaluation approaches have their roots in critical success factors and are thus listed here as well. As with individual enterprise architecture metrics other than maturity, the IT side of enterprise architecture dominates this category.

Aivosto Oy. (2006a). "Miscellaneous Object-oriented Metrics." from http://www.aivosto.com/project/help/pm-oo-misc.html.

Aivosto Oy. (2006b). "MOOD Metrics." from http://www.aivosto.com/project/help/pm-oo-mood.html.

Al-Mashari, M. and M. Zairi (1999). "BPR implementation process: an analysis of key success and failure factors." Business Process Management Journal 5(1): 87-112.

Altizer, B. (2004). "Measurements and Checkpoints (a presentation)." Retrieved September 22, 2006.

Aoyama, M. (2002). Metrics and analysis of software architecture evolution with discontinuity. International Conference on Software Engineering. Proceedings of the International Workshop on Principles of Software Evolution, Orlando, Florida, ACM Press.

Avritzer, A. and E. J. Weyuker (1998). Investigating Metrics for Architectural Assessment. Proceedings of the Fifth International Software Metrics Symposium, Metrics 1998. Bethesda, MD, IEEE Computer Society: 4-10.

Avritzer, A. and E. J. Weyuker (1999). "Metrics to Assess the Likelihood of Project Success Based on Architecture Reviews." Empirical Software Engineering 4(3): 199 - 215.

Aziz, S., T. Obitz, R. Modi and S. Sarkar. (2006). "Enterprise Architecture: A Governance Framework - Part II: Making Enterprise Architecture Work within the Organization." Infosys White Paper Retrieved 22 August, 2006, from http://www.infosys.com/enterprise-architecture/.

Bache, R. B., G. (1994). Software Metrics for Product Assessment., McGraw-Hill.

Baker, D. C. and M. Janiszewski. (2005). "7 Essential Elements of EA." Enterprise Architect Retrieved 17.6.2005, from http://www.ftponline.com/ea/magazine/summer2005/features/dbaker/.

Barbacci, M. R., M. H. Klein and C. B. Weinstock (1997). Principles for Evaluating the Quality Attributes of a Software Architecture, Technical Report CMU/SEI-96-TR-036, Software Engineering Institute, Carnegie Mellon University.

Barber, K. S. G., T.J. (2000). Tool support for systematic class identification in object-oriented software architectures. Proceedings. 37th International Conference on Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Sydney, NSW.

Barnett, L. (2005). Metrics for Application Development. Best Practices, Forrester Research.

Belle, J.-P. V. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. The 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries: 210-215.

Bitterman, M. (2004). "IT Metrics for the Information Age." from http://www.performance-measurement.net.

Boehm, B. (1994). Software architectures: critical success factors and cost drivers. Proceedings of the 16th International Conference on Software Engineering. Sorrento, Italy, IEEE Computer Society: 365.

Bredemeyer Consulting. (2000). "Software Architecting Success Factors and Pitfalls." from http://www.bredemeyer.com/CSFs_pitfalls.htm.

Brown, C. V. (2003). "Performance Metrics for IT Human Resource Alignment." Information Systems Management 20(4): 36-42.

Carbone, J. (2004). IT Architecture Toolkit. Upper Saddle River, USA, Prentice Hall.

Carbone, J. (2005). "Ten Signs of Enterprise Architecture Maturity." 2005, from http://www.harriskern.com/index.php?m=p&pid=377&authorid=&aid=190.

Chan, Y. E., S. L. Huff, D. W. Barclay and D. G. Copeland (1997). "Business Strategic Orientation, Information Systems Strategic Orientation, and Strategic Alignment." Information Systems Research 8(2): 125-150.

Chidamber, S. R. K., C.F. (1994). "A metrics suite for object oriented design." IEEE Transactions on Software Engineering 20(6): 476 - 493.

Chulani, S. R., B.; Santhanam, P.; Leszkowicz, R. (2003). Metrics for managing customer view of software quality. Proceedings. Ninth International Software Metrics Symposium, 2003.

Claxton, J. C. and P. A. McDougall (2000) "Measuring the Quality of Models." The Data Administration Newsletter (TDAN.com) Volume, DOI:

Crow, K. "Product Development Metrics List

http://npd-solutions.com/metrics.html."

Curran, C. (2005). "Link IT Investments to Business Metrics." Enterprise Architect 3(1): 16-18.

Dias, O. P., I. C. Teixeira and J. P. Teixeira (1999). "Metrics and Criteria for Quality Assessment of Testable Hw/Sw Systems Architectures." Journal of Electronic Testing: Theory and Applications 14: 149-158.

Dikel, D., D. Kane, S. Ornburn and J. Wilson. (1995). "Software Architecture Case Study: Organizational Success Factors." from http://www.vraps.com/files/archcase_exec_sum.doc http://www.vraps.com/files/.

DoC (2003). IT Architecture Capability Maturity Model. Department of Commerce (DoC).

Drudis, A. and J. Mendonca. "The Architecture of Web Applications." Retrieved 26.5.2005, from http://www.interex.org/pubcontent/enterprise/may01/14drudis.html.

Drury, C. (1992). Management and Cost Accounting. London, UK, Chapman & Hall.

El Emam, K. (2001). A primer on object-oriented measurement. Proceedings. Seventh International Software Metrics Symposium, METRICS 2001. London.

ETS Office (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

Eurocontrol. (2006). "WP 8.1.1 – Define Methodology For Validation Within OATA. Architecture Compliance Assessment Process. Edition 2.0." 2.0. from http://www.eurocontrol.int/valug/gallery/content/public/OATA-P2-D8.1.1-01%20DMVO%20Architecture%20Compliance%20Assessment%20Process.doc.

Fabbrini, F., M. Fusani and S. Gnesi (1998). Quality Evaluation based on Architecture Analysis. Proceedings of the International Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA'98). Marsala, Italy.

Fenton, N. (1991). Software Metrics - a rigorous approach, Chapman & Hall.

GAO. (2003a). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." from http://www.gao.gov/new.items/d03584g.pdf.

GAO (2003b). Leadership Remains Key to Agencies Making Progress on Enterprise Architecture Efforts, General Accounting Office (GAO).

Government Accountability Office (GAO). (2003). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." 2004.

Grady, R. B. (1992). Practical Software Metrics for Project Management and Process Improvement, Prentice Hall.

Gunasekaran, A., C. Patel and E. Tirtiroglu (2001). "Performance measures and metrics in a supply chain environment." International Journal of Operations & Production Management 21(1/2): 71-87.

Gustafsson, J., J. Paakki, L. Nenonen and A. I. Verkamo (2002). Architecture-Centric Software Evolution by Software Metrics and Design Patterns. Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02). Budapest, IEEE Computer Society: 108-115.

Haley, G. (2005). "Software Architecture Evaluation - Transforming a craft into a business process." Retrieved 26.5., 2005, from http://sunset.usc.edu/gsaw/gsaw2005/s9b/haley.pdf.

Halley, M. R. B., C. (2005). Enterprise transformation to a service oriented architecture: successful patterns. Proceedings. 2005 IEEE International Conference on Web Services, ICWS 2005.

Hass, A. M. J. J., J.; Pries-Heje, J. (1998). Does ISO 9001 increase software development maturity? Proceedings. 24th Euromicro Conference, 1998. Vasteras.

Hazra, T. K. (2005). "Getting Your EA Metrics Right (a presentation)." 2005.

Henderson, J. and N. Venkatraman (1992). "Aligning Business and Information Technology Domains: Strategic Planning in Hospitals." Hospital and Health Services Administrative 37(1): 71-87.

Hirschheim, R. and R. Sabherwal (2001). "Detours in the Path toward Strategic Information Systems Alignment." California Management Review 44(1): 86-108.

Hirvonen, A. (2005). Enterprise Architecture Planning in Practice: The Perspectives of Information and Communication Technology Service Provider and End-user. Department of Computer Science and Information Systems. Jyväskylä, University of Jyväskylä.

IEEE (1998). IEEE Standard for a Software Quality Metrics Methodology, IEEE Std 1061-1998.

Intel (2003). Managing IT Investments - Intel's IT Business Value metrics program, Intel Technology.

Ionita, M. T., P. America, D. K. Hammer, H. Obbink and J. J. M. Trienekens (2004). A scenario-driven approach for value, risk, and cost analysis in system architecting for innovation. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

ISO (2003a). ISO/IEC TR 9126-2:2003, Software engineering -- Product quality -- Part 2: External metrics, ISO.

ISO (2003b). ISO/IEC TR 9126-3:2003, Software engineering -- Product quality -- Part 3: Internal metrics, ISO.

Ivkovic, I. K., K. (2006). A Framework for Software Architecture Refactoring using Model Transformations and Semantic Annotations. Proceedings of the 10th European Conference on Software Maintenance and Reengineering, CSMR 2006.

Kan, S. H. (2005). Metrics and Models in Software Quality Engineering, Addison-Wesley.

Kazman, R. (1996). Tool Support for Architecture Analysis and Design. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, California, United States: 94-97.

Kornecki, A. J. and J. Zalewski. (2004). "TUTORIAL: Software Development for Real-Time Safety-Critical Applications." from http://ieeexplore.ieee.org/iel5/9714/30657/01416838.pdf?arnumber=1416838.

Krikhaar, R. P., A.; Sellink, A.; Stroucken, M.; Verhoef, C. (1999). A two-phase process for software architecture improvement. Proceedings. IEEE International Conference on Software Maintenance, 1999. (ICSM '99).

Krueger, I. H. (2004). Evaluating Software Architectures: Stakeholders, Metrics, Results, Migration Strategies. Department of Computer Science & Engineering. San Diego, University of California.

Kulpa, M. K. and K. A. Johnson (2003). Interpreting the CMMI: A Process Improvement Approach, Auerbach Publications.

Langelier, G., H. Sahraoui and P. Poulin (2005). Visualization-based analysis of quality for large-scale software systems. Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. Long Beach, CA, USA, ACM Press.

Lindvall, M., R. Tesoriero and P. Costa (2002). Avoiding Architectural Degeneration: An Evaluation Process for Software Architecture. Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS'02), IEEE Computer Society: 77-86.

Lindvall, M., R. T. Tvedt and P. Costa (2003). "An Empirically-Based Process for Software Architecture Evaluation." Empirical Software Engineering 8(1): 83-108.

Livingston, J., K. Prosise and R. J. Altizer (1995). Process Improvement Matrix: A Tool for Measuring Progress Towards Better Quality. Proceedings of 5th International Conference on Software Quality. Austin, Texas.

Lopez, M. (2000). An Evaluation Theory Perspective of the Architecture Tradeoff Analysis Method (ATAM). Pittsburg, USA, The Software Engineering Institute, Carnegie Mellon University.

Lorenz, M. and J. Kidd (1994). Object-oriented software metrics, PTR Prentice Hall, Englewood Cliffs, New Jersey.

Losavio, F., L. Chirinos, N. Lévy and A. Ramdane-Cherif (2003). "Quality Characteristics for Software Architecture." Journal of Object Technology 2(2): 133-150.

Losavio, F., L. Chirinos, A. Matteo, N. Lévy and A. Ramdane-Cherif (2004). "ISO Quality Standards for Measuring Architectures." The Journal of Systems and Software 72: 209-223.

Losavio, F., A. Matteo, J. O. Ordaz, N. Levy and R. Marcano-Kamenoff (2002). Quality Characteristics to Select an Architecture for Real-time Internet Applications. 4th International Software Quality Week Europe (QWE2000). Brussels, Belgium.

Luftman, J. (2003). "Assessing IT/Business Alignment." Information Systems Management 20(4): 9-15.

Luftman, J. N., R. Papp and T. Brier (1999). "Enablers and Inhibitors of Business-IT Alignment." Communications of the Association for Information Systems 1(11).

Lung, C.-H. and K. Kalaichelvan (2000). "An Approach to Quantitative Software Architecture Sensitivity Analysis." International Journal of Software Engineering and Knowledge Engineering 10(1): 97-114.

Matinlassi, M. (2004). Evaluating the portability and maintainability of software product family architecture: terminal software case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Matinlassi, M., E. Niemelä and L. Dobrica (2002). Quality-Driven Architecture Design and Quality Analysis Method - A Revolutionary Initiation Approach to a Product Line Architecture. VTT Publications. Espoo, Technical Research Centre of Finland, VTT: 128.

McCay, B. M. (1996). Some Thoughts on the Quality of a Computer-Based System's Architecture. Engineering of the IEEE Symposium and Workshop on Computer-Based Systems. Friedrichshafen, IEEE Computer Society: 228-234.

META Group Inc. (2000a). "Architecture Capability Assessment." META Practice 4(7).

META Group Inc. (2000b). "Architecture Maturity Assessment." METAView Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0378/mv0378.html.

META Group Inc. (2002). "Enterprise Architecture Maturity." METAView, Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0599/mv0599.html.

META Group Inc. (2004a). "Architecture Program Maturity Assessment: Findings and Trends." Enterprise Planning & Architecture Strategies, Teleconference 2118, 23 September 2004, from http://www.metagroup.com/us/displayArticle.do?oid=49449.

META Group Inc. (2004b). "Planning the Enterprise Architecture Measurement Program." META Group Research, Enterprise Planning & Architecture Strategies, Practice Summary, from http://www.metagroup.com/us/displayArticle.do?oid=50037.

Morganwalp, J. M. and A. P. Sage (2004). "Enterprise Architecture Measures of Effectiveness." International Journal of Technology, Policy and Management 4(1): 81-94.

Niemi, E. (2006). "Enterprise Architecture Work Overview in Three Finnish Business Enterprises." WSEAS Transactions on Business and Economics 3(9): 628-635.

Ogasawara, H. Y., A.; Kojo, M. (1996). Experiences of software quality management using metrics through the life-cycle. Proceedings of the 18th International Conference on Software Engineering. Berlin.

OMB (2005a). Federal Enterprise Architecture Program EA Assessment Framework 2.0, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB (2005b). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB FEA Program Management Office (2005). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

Perkins, A. (2003). "Critical Success Factors for Enterprise Architecture Engineering (Visible Solutions Whitepaper)." from www.visible.com.

Potter, N. S. and M. E. Sakry (2002). Making Process Improvement Work: A Concise Action Guide for Software Managers and Practitioners, Addison Wesley.

Purhonen, A. (2004). Performance optimization of embedded software architecture - a case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Ramsay, P. (2004). "Ensuring that Architecture Works for the Enterprise." Executive Reports, Cutter Consortium 7(13).

Rehkopf, T. W. and N. Wybolt (2003). "Top 10 Architecture Land Mines." IT Professional 5(6): 36-43.

Rico, D. F. (2004). ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers, J. Ross Publishing.

Rico, D. F. (2005). "A Framework for Measuring the ROI of Enterprise Architecture."

Rosenfeld, L., K. Instone, S. Cupala, J. J. Garrett, M. Hearst, G. Marchionini and N. Ragouzis (2001). Measuring Information Architecture Quality: Prove It (or not)! CHI '01 extended abstracts on Human factors in computing systems. Seattle, Washington, USA, ACM Press: 219-220.

Rosser, B. (2006). Measuring the Value of Enterprise Architecture: Metrics and ROI, Gartner.

Scallon, G. (1999). "Model Based Development." Retrieved 26.5.2005, from http://www.stc-online.org/cd-rom/1999/slides/MBDslid1.pdf.

Schekkerman, J. (2003). "Extended Enterprise Maturity Model (E2AMM)." from http://www.enterprise-architecture.info/Images/E2AF/E2AMMv2.PDF.

Schekkerman, J. (2004). How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework. Canada, Trafford.

Sharareh, A., G. Matteo and T. Gianluca (2002). Quantitative analysys for telecom/datacom software architecture. Proceedings of the 3rd international workshop on Software and performance WOSP '02, Rome, Italy, ACM Press.

Shereshevsky, M., H. Ammari, N. Gradetsky, A. Mili and H. H. Ammar (2001). Information Theoretic Metrics for Software Architectures. Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC 2001), Chicago, IL, USA, IEEE Computer Society.

SMLab. (2006). "Software Metrics Classification." from http://irb.cs.uni-magdeburg.de/sw-eng/us/metclas/index.shtml.

Snedaker, S. (2005). How to Cheat at IT Project Management, Syngress.

Tahvildari, L., K. Kontogiannis and J. Mylopoulos (2003). "Quality-Driven Software Re-engineering." The Journal of Systems and Software 66: 225-239.

Teo, T. S. H. and J. S. K. Ang (1999). "Critical success factors in the alignment of IS plans with business plans." International Journal of Information Management 19(2): 173-185.

The Office of Enterprise Technology Strategies (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

Tvedt, R. T., M. Lindvall and P. Costa (2002). A Process for Software Architecture Evaluation Using Metrics. Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02): 191-196.

Van Belle, J. P. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. Proceedings of the 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (SAICSIT-2004): 210-215.

van der Raadt, B., J. Soetendal, M. Perdeck and H. v. Vliet (2004). Polyphony in Architecture. Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), IEEE Computer Society: 533-542.

Wang, T., A. Hassan, A. Guedem, W. Abdelmoez, K. Goseva-Popstojanova and H. Ammar (2003). Architectural level risk assessment tool based on UML specifications. Proceedings of the 25th International Conference on Software Engineering. Portland, Oregon, IEEE Computer Society.

Whitman, L., K. Ramachandran and V. Ketkar (2001). A Taxonomy of a Living Model of the Enterprise. Proceedings of the 2001 Winter Simulation Conference. B. A. Peters, Smith, J.S., Medeiros, D.J., Rohrer, M.W. Arlington, Virginia, IEEE Computer Society: 848-855.

Yacoub, S. M. A., H.H. (2002). "A methodology for architecture-level reliability risk analysis." IEEE Transactions on Software Engineering 28(6): 529 - 547.

Ylimäki, T. (2005). Towards Critical Success Factors for Enterprise Architecture. AISA Project report, Information Technology Research Institute, University of Jyväskylä.

Ylimäki, T. (2006a). Assessing Architectural Work - Criteria and Metrics for Evaluating Communication & Common Language and Commitment. AISA Project Report. Jyväskylä, Finland, Information Technology Research Institute, University of Jyväskylä.

Ylimäki, T. (2006b). "Potential Critical Success Factors for Enterprise Architecture." Journal of Enterprise Architecture 2(4): 29-40.

Zalewski, J. (2005). "From software sensitivity to software dynamics: performance metrics for real-time software architectures." ACM SIGBED Review 2(3): 20 - 24.

Zhao, J. (1998). On assessing the complexity of software architectures. Proceedings of the third international workshop on Software architecture, Orlando, Florida, United States, ACM Press.

1.5 Software Architecture Success Factors, Evaluation Criteria and Metrics

Like the previous category, this category includes similar references in the software architecture domain. Some references discuss software architecture critical success factors, but mostly the focus is on software architecture metrics and quality attributes. Moreover, metrics related to software engineering are introduced in a number of references. Some measurement approaches are also introduced, with several references listed already in the previous category, indicating approaches that can be used in or cover both domains.

Aivosto Oy. (2006a). "Miscellaneous Object-oriented Metrics." from http://www.aivosto.com/project/help/pm-oo-misc.html.

Aivosto Oy. (2006b). "MOOD Metrics." from http://www.aivosto.com/project/help/pm-oo-mood.html.

Al-Mashari, M. and M. Zairi (1999). "BPR implementation process: an analysis of key success and failure factors." Business Process Management Journal 5(1): 87-112.

Altizer, B. (2004). "Measurements and Checkpoints (a presentation)." Retrieved September 22, 2006.

Aoyama, M. (2002). Metrics and analysis of software architecture evolution with discontinuity. International Conference on Software Engineering. Proceedings of the International Workshop on Principles of Software Evolution, Orlando, Florida, ACM Press.

Avritzer, A. and E. J. Weyuker (1998). Investigating Metrics for Architectural Assessment. Proceedings of the Fifth International Software Metrics Symposium, Metrics 1998. Bethesda, MD, IEEE Computer Society: 4-10.

Avritzer, A. and E. J. Weyuker (1999). "Metrics to Assess the Likelihood of Project Success Based on Architecture Reviews." Empirical Software Engineering 4(3): 199 - 215.

Aziz, S., T. Obitz, R. Modi and S. Sarkar. (2006). "Enterprise Architecture: A Governance Framework - Part II: Making Enterprise Architecture Work within the Organization." Infosys White Paper Retrieved 22 August, 2006, from http://www.infosys.com/enterprise-architecture/.

Bache, R. B., G. (1994). Software Metrics for Product Assessment., McGraw-Hill.

Baker, D. C. and M. Janiszewski. (2005). "7 Essential Elements of EA." Enterprise Architect Retrieved 17.6.2005, from http://www.ftponline.com/ea/magazine/summer2005/features/dbaker/.

Barbacci, M. R., M. H. Klein and C. B. Weinstock (1997). Principles for Evaluating the Quality Attributes of a Software Architecture, Technical Report CMU/SEI-96-TR-036, Software Engineering Institute, Carnegie Mellon University.

Barber, K. S. G., T.J. (2000). Tool support for systematic class identification in object-oriented software architectures. Proceedings. 37th International Conference on Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Sydney, NSW.

Barnett, L. (2005). Metrics for Application Development. Best Practices, Forrester Research.

Belle, J.-P. V. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. The 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries: 210-215.

Bitterman, M. (2004). "IT Metrics for the Information Age." from http://www.performance-measurement.net.

Boehm, B. (1994). Software architectures: critical success factors and cost drivers. Proceedings of the 16th International Conference on Software Engineering. Sorrento, Italy, IEEE Computer Society: 365.

Bredemeyer Consulting. (2000). "Software Architecting Success Factors and Pitfalls." from http://www.bredemeyer.com/CSFs_pitfalls.htm.

Brown, C. V. (2003). "Performance Metrics for IT Human Resource Alignment." Information Systems Management 20(4): 36-42.

Carbone, J. (2004). IT Architecture Toolkit. Upper Saddle River, USA, Prentice Hall.

Carbone, J. (2005). "Ten Signs of Enterprise Architecture Maturity." 2005, from http://www.harriskern.com/index.php?m=p&pid=377&authorid=&aid=190.

Chan, Y. E., S. L. Huff, D. W. Barclay and D. G. Copeland (1997). "Business Strategic Orientation, Information Systems Strategic Orientation, and Strategic Alignment." Information Systems Research 8(2): 125-150.

Chidamber, S. R. K., C.F. (1994). "A metrics suite for object oriented design." IEEE Transactions on Software Engineering 20(6): 476 - 493.

Chulani, S. R., B.; Santhanam, P.; Leszkowicz, R. (2003). Metrics for managing customer view of software quality. Proceedings. Ninth International Software Metrics Symposium, 2003.

Claxton, J. C. and P. A. McDougall (2000) "Measuring the Quality of Models." The Data Administration Newsletter (TDAN.com) Volume, DOI:

Crow, K. "Product Development Metrics List
http://npd-solutions.com/metrics.html."

Curran, C. (2005). "Link IT Investments to Business Metrics." Enterprise Architect 3(1): 16-18.

Dias, O. P., I. C. Teixeira and J. P. Teixeira (1999). "Metrics and Criteria for Quality Assessment of Testable Hw/Sw Systems Architectures." Journal of Electronic Testing: Theory and Applications 14: 149-158.

Dikel, D., D. Kane, S. Ornburn and J. Wilson. (1995). "Software Architecture Case Study: Organizational Success Factors." from http://www.vraps.com/files/archcase_exec_sum.doc http://www.vraps.com/files/.

DoC (2003). IT Architecture Capability Maturity Model. Department of Commerce (DoC).

Drudis, A. and J. Mendonca. "The Architecture of Web Applications." Retrieved 26.5.2005, from http://www.interex.org/pubcontent/enterprise/may01/14drudis.html.

Drury, C. (1992). Management and Cost Accounting. London, UK, Chapman & Hall.

El Emam, K. (2001). A primer on object-oriented measurement. Proceedings. Seventh International Software Metrics Symposium, METRICS 2001. London.

ETS Office (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

Eurocontrol. (2006). "WP 8.1.1 – Define Methodology For Validation Within OATA. Architecture Compliance Assessment Process. Edition 2.0." 2.0. from http://www.eurocontrol.int/valug/gallery/content/public/OATA-P2-D8.1.1-01%20DMVO%20Architecture%20Compliance%20Assessment%20Process.doc.

Fabbrini, F., M. Fusani and S. Gnesi (1998). Quality Evaluation based on Architecture Analysis. Proceedings of the International Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA'98). Marsala, Italy.

Fenton, N. (1991). Software Metrics - a rigorous approach, Chapman & Hall.

GAO. (2003a). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." from http://www.gao.gov/new.items/d03584g.pdf.

GAO (2003b). Leadership Remains Key to Agencies Making Progress on Enterprise Architecture Efforts, General Accounting Office (GAO).

Government Accountability Office (GAO). (2003). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." 2004.

Grady, R. B. (1992). Practical Software Metrics for Project Management and Process Improvement, Prentice Hall.

Gunasekaran, A., C. Patel and E. Tirtiroglu (2001). "Performance measures and metrics in a supply chain environment." International Journal of Operations & Production Management 21(1/2): 71-87.

Gustafsson, J., J. Paakki, L. Nenonen and A. I. Verkamo (2002). Architecture-Centric Software Evolution by Software Metrics and Design Patterns. Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02). Budapest, IEEE Computer Society: 108-115.

Haley, G. (2005). "Software Architecture Evaluation - Transforming a craft into a business process." Retrieved 26.5., 2005, from http://sunset.usc.edu/gsaw/gsaw2005/s9b/haley.pdf.

Hass, A. M. J. J., J.; Pries-Heje, J. (1998). Does ISO 9001 increase software development maturity? Proceedings. 24th Euromicro Conference, 1998. Vasteras.

Hazra, T. K. (2005). "Getting Your EA Metrics Right (a presentation)." 2005.

Henderson, J. and N. Venkatraman (1992). "Aligning Business and Information Technology Domains: Strategic Planning in Hospitals." Hospital and Health Services Administrative 37(1): 71-87.

IEEE (1998). IEEE Standard for a Software Quality Metrics Methodology, IEEE Std 1061-1998.

Intel (2003). Managing IT Investments - Intel's IT Business Value metrics program, Intel Technology.

Ionita, M. T., P. America, D. K. Hammer, H. Obbink and J. J. M. Trienekens (2004). A scenario-driven approach for value, risk, and cost analysis in system architecting for innovation. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

ISO (2003a). ISO/IEC TR 9126-2:2003, Software engineering -- Product quality -- Part 2: External metrics, ISO.

ISO (2003b). ISO/IEC TR 9126-3:2003, Software engineering -- Product quality -- Part 3: Internal metrics, ISO.

Ivkovic, I. K., K. (2006). A Framework for Software Architecture Refactoring using Model Transformations and Semantic Annotations. Proceedings of the 10th European Conference on Software Maintenance and Reengineering, CSMR 2006.

Kan, S. H. (2005). Metrics and Models in Software Quality Engineering, Addison-Wesley.

Kazman, R. (1996). Tool Support for Architecture Analysis and Design. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, California, United States: 94-97.

Kornecki, A. J. and J. Zalewski. (2004). "TUTORIAL: Software Development for Real-Time Safety-Critical Applications." from http://ieeexplore.ieee.org/iel5/9714/30657/01416838.pdf?arnumber=1416838.

Krikhaar, R. P., A.; Sellink, A.; Stroucken, M.; Verhoef, C. (1999). A two-phase process for software architecture improvement. Proceedings. IEEE International Conference on Software Maintenance, 1999. (ICSM '99).

Krueger, I. H. (2004). Evaluating Software Architectures: Stakeholders, Metrics, Results, Migration Strategies. Department of Computer Science & Engineering. San Diego, University of California.

Kulpa, M. K. and K. A. Johnson (2003). Interpreting the CMMI: A Process Improvement Approach, Auerbach Publications.

Langelier, G., H. Sahraoui and P. Poulin (2005). Visualization-based analysis of quality for large-scale software systems. Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering. Long Beach, CA, USA, ACM Press.

Lindvall, M., R. Tesoriero and P. Costa (2002). Avoiding Architectural Degeneration: An Evaluation Process for Software Architecture. Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS'02), IEEE Computer Society: 77-86.

Lindvall, M., R. T. Tvedt and P. Costa (2003). "An Empirically-Based Process for Software Architecture Evaluation." Empirical Software Engineering 8(1): 83-108.

Livingston, J., K. Prosise and R. J. Altizer (1995). Process Improvement Matrix: A Tool for Measuring Progress Towards Better Quality. Proceedings of 5th International Conference on Software Quality. Austin, Texas.

Lopez, M. (2000). An Evaluation Theory Perspective of the Architecture Tradeoff Analysis Method (ATAM). Pittsburg, USA, The Software Engineering Institute, Carnegie Mellon University.

Lorenz, M. and J. Kidd (1994). Object-oriented software metrics, PTR Prentice Hall, Englewood Cliffs, New Jersey.

Losavio, F., L. Chirinos, N. Lévy and A. Ramdane-Cherif (2003). "Quality Characteristics for Software Architecture." Journal of Object Technology 2(2): 133-150.

Losavio, F., L. Chirinos, A. Matteo, N. Lévy and A. Ramdane-Cherif (2004). "ISO Quality Standards for Measuring Architectures." The Journal of Systems and Software 72: 209-223.

Losavio, F., A. Matteo, J. O. Ordaz, N. Levy and R. Marcano-Kamenoff (2002). Quality Characteristics to Select an Architecture for Real-time Internet Applications. 4th International Software Quality Week Europe (QWE2000). Brussels, Belgium.

Luftman, J. (2003). "Assessing IT/Business Alignment." Information Systems Management 20(4): 9-15.

Luftman, J. N., R. Papp and T. Brier (1999). "Enablers and Inhibitors of Business-IT Alignment." Communications of the Association for Information Systems 1(11).

Lung, C.-H. and K. Kalaichelvan (2000). "An Approach to Quantitative Software Architecture Sensitivity Analysis." International Journal of Software Engineering and Knowledge Engineering 10(1): 97-114.

Matinlassi, M. (2004). Evaluating the portability and maintainability of software product family architecture: terminal software case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Matinlassi, M., E. Niemelä and L. Dobrica (2002). Quality-Driven Architecture Design and Quality Analysis Method - A Revolutionary Initiation Approach to a Product Line Architecture. VTT Publications. Espoo, Technical Research Centre of Finland, VTT: 128.

McCay, B. M. (1996). Some Thoughts on the Quality of a Computer-Based System's Architecture. Engineering of the IEEE Symposium and Workshop on Computer-Based Systems. Friedrichshafen, IEEE Computer Society: 228-234.

META Group Inc. (2000a). "Architecture Capability Assessment." META Practice 4(7).

META Group Inc. (2000b). "Architecture Maturity Assessment." METAView Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0378/mv0378.html.

META Group Inc. (2002). "Enterprise Architecture Maturity." METAView, Daily Audio Briefing, from http://www.metagroup.com/metaview/mv0599/mv0599.html.

META Group Inc. (2004a). "Architecture Program Maturity Assessment: Findings and Trends." Enterprise Planning & Architecture Strategies, Teleconference 2118, 23 September 2004, from http://www.metagroup.com/us/displayArticle.do?oid=49449.

META Group Inc. (2004b). "Planning the Enterprise Architecture Measurement Program." META Group Research, Enterprise Planning & Architecture Strategies, Practice Summary, from http://www.metagroup.com/us/displayArticle.do?oid=50037.

Morganwalp, J. M. and A. P. Sage (2004). "Enterprise Architecture Measures of Effectiveness." International Journal of Technology, Policy and Management 4(1): 81-94.

Niemi, E. (2006). "Enterprise Architecture Work Overview in Three Finnish Business Enterprises." WSEAS Transactions on Business and Economics 3(9): 628-635.

Ogasawara, H. Y., A.; Kojo, M. (1996). Experiences of software quality management using metrics through the life-cycle. Proceedings of the 18th International Conference on Software Engineering. Berlin.

OMB (2005a). Federal Enterprise Architecture Program EA Assessment Framework 2.0, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB (2005b). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB FEA Program Management Office (2005). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

Potter, N. S. and M. E. Sakry (2002). Making Process Improvement Work: A Concise Action Guide for Software Managers and Practitioners, Addison Wesley.

Purhonen, A. (2004). Performance optimization of embedded software architecture - a case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Rehkopf, T. W. and N. Wybolt (2003). "Top 10 Architecture Land Mines." IT Professional 5(6): 36-43.

Rico, D. F. (2004). ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers, J. Ross Publishing.

Rico, D. F. (2005). "A Framework for Measuring the ROI of Enterprise Architecture."

Rosenfeld, L., K. Instone, S. Cupala, J. J. Garrett, M. Hearst, G. Marchionini and N. Ragouzis (2001). Measuring Information Architecture Quality: Prove It (or not)! CHI '01 extended abstracts on Human factors in computing systems. Seattle, Washington, USA, ACM Press: 219-220.

Rosser, B. (2006). Measuring the Value of Enterprise Architecture: Metrics and ROI, Gartner.

Scallon, G. (1999). "Model Based Development." Retrieved 26.5.2005, from http://www.stc-online.org/cd-rom/1999/slides/MBDslid1.pdf.

Schekkerman, J. (2003). "Extended Enterprise Maturity Model (E2AMM)." from http://www.enterprise-architecture.info/Images/E2AF/E2AMMv2.PDF.

Schekkerman, J. (2004). How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework. Canada, Trafford.

Schekkerman, J. (2004b). How to survive in the jungle of Enterprise Architecture Frameworks - Creating or choosing an Enterprise Architecture Framework. Canada, Trafford.

Sharareh, A., G. Matteo and T. Gianluca (2002). Quantitative analysys for telecom/datacom software architecture. Proceedings of the 3rd international workshop on Software and performance WOSP '02, Rome, Italy, ACM Press.

Shereshevsky, M., H. Ammari, N. Gradetsky, A. Mili and H. H. Ammar (2001). Information Theoretic Metrics for Software Architectures. Proceedings of the 25th Annual International Computer Software and Applications Conference (COMPSAC 2001), Chicago, IL, USA, IEEE Computer Society.

SMLab. (2006). "Software Metrics Classification." from http://irb.cs.uni-magdeburg.de/sw-eng/us/metclas/index.shtml.

Snedaker, S. (2005). How to Cheat at IT Project Management, Syngress.

Tahvildari, L., K. Kontogiannis and J. Mylopoulos (2003). "Quality-Driven Software Re-engineering." The Journal of Systems and Software 66: 225-239.

Teo, T. S. H. and J. S. K. Ang (1999). "Critical success factors in the alignment of IS plans with business plans." International Journal of Information Management 19(2): 173-185.

The Office of Enterprise Technology Strategies (2003). Maturity Review Plan (Version 1.0.0), Office of Enterprise Technology Strategies, State of North Carolina.

Tvedt, R. T., M. Lindvall and P. Costa (2002). A Process for Software Architecture Evaluation Using Metrics. Proceedings of the 27th Annual NASA Goddard/IEEE Software Engineering Workshop (SEW-27'02): 191-196.

Van Belle, J. P. (2004). A Proposed Framework for the Analysis and Evaluation of Business Models. Proceedings of the 2004 Annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (SAICSIT-2004): 210-215.

van der Raadt, B., J. Soetendal, M. Perdeck and H. v. Vliet (2004). Polyphony in Architecture. Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), IEEE Computer Society: 533-542.

Wang, T., A. Hassan, A. Guedem, W. Abdelmoez, K. Goseva-Popstojanova and H. Ammar (2003). Architectural level risk assessment tool based on UML specifications. Proceedings of the 25th International Conference on Software Engineering. Portland, Oregon, IEEE Computer Society.

Whitman, L., K. Ramachandran and V. Ketkar (2001). A Taxonomy of a Living Model of the Enterprise. Proceedings of the 2001 Winter Simulation Conference. B. A. Peters, Smith, J.S., Medeiros, D.J., Rohrer, M.W. Arlington, Virginia, IEEE Computer Society: 848-855.

Yacoub, S. M. A., H.H. (2002). "A methodology for architecture-level reliability risk analysis." IEEE Transactions on Software Engineering 28(6): 529 - 547.

Ylimäki, T. (2006a). Assessing Architectural Work - Criteria and Metrics for Evaluating Communication & Common Language and Commitment. AISA Project Report. Jyväskylä, Finland, Information Technology Research Institute, University of Jyväskylä.

Ylimäki, T. (2006b). "Potential Critical Success Factors for Enterprise Architecture." Journal of Enterprise Architecture 2(4): 29-40.

Zalewski, J. (2005). "From software sensitivity to software dynamics: performance metrics for real-time software architectures." ACM SIGBED Review 2(3): 20 - 24.

Zhao, J. (1998). On assessing the complexity of software architectures. Proceedings of the third international workshop on Software architecture, Orlando, Florida, United States, ACM Press.

2 ARCHITECTURE EVALUATION METHODS

2.1 Enterprise architecture evaluation methods

This category concentrates on enterprise architecture evaluation methods. Most of the references have been listed in the enterprise architecture metrics domain already – usually the methods and metrics are interrelated. However, the references listed here describe at least one evaluation method for a part of enterprise architecture, not solely metrics. Maturity evaluation methods could have been listed here as well, but we consider them to be related more to enterprise architecture critical success factors. It should be noted that methods for evaluating enterprise architecture as a whole do not yet exist; methods should be adopted and adapted from e.g. the software architecture domain and combined to evaluate the entire enterprise architecture. A description of methods for evaluating the viewpoints of enterprise architecture is presented in AISA project report Architecture Evaluation Methods (Hoffmann 2007).

Aziz, S., T. Obitz, R. Modi and S. Sarkar. (2006). "Enterprise Architecture: A Governance Framework - Part II: Making Enterprise Architecture Work within the Organization." Infosys White Paper Retrieved 22 August, 2006, from http://www.infosys.com/enterprise-architecture/.

Dunsire, K., T. O'Neill, M. Denford and J. Leaney (2005). The ABACUS architectural approach to computer-based system and enterprise evolution. 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS '05), IEEE Computer Society.

Erder, M. and P. Pureur (2003). "QFD in the Architecture Development Process." IT Professional 5(6): 44-52.

Eurocontrol. (2006). "WP 8.1.1 – Define Methodology For Validation Within OATA. Architecture Compliance Assessment Process. Edition 2.0." 2.0. from http://www.eurocontrol.int/valug/gallery/content/public/OATA-P2-D8.1.1-01%20DMVO%20Architecture%20Compliance%20Assessment%20Process.doc.

Gartner (2002). Return on Enterprise Architecture: Measure It in Asset Productivity. GartnerG2 Report. Stamford, USA, Gartner, Inc.

Hirvonen, A. (2005). Enterprise Architecture Planning in Practice: The Perspectives of Information and Communication Technology Service Provider and End-user. Department of Computer Science and Information Systems. Jyväskylä, University of Jyväskylä.

Kluge, C., A. Dietzsch and M. Rosemann (2006). How to Realize Corporate Value from Enterprise Architecture. the Proceedings of the 14th European Conference on Information Systems (ECIS 2006). Göteborg, Sweden, Association for Information Systems.

Lee, S. M., D. L. Olson, S. Trimi and K. M. Rosacker (2005). "An Integrated Method to Evaluate Business Process Alternatives." Business Process Management Journal 11(2): 198-212.

Morganwalp, J. M. and A. P. Sage (2004). "Enterprise Architecture Measures of Effectiveness." International Journal of Technology, Policy and Management 4(1): 81-94.

Rosenfeld, L., K. Instone, S. Cupala, J. J. Garrett, M. Hearst, G. Marchionini and N. Ragouzis (2001). Measuring Information Architecture Quality: Prove It (or not)! CHI '01 extended abstracts on Human factors in computing systems. Seattle, Washington, USA, ACM Press: 219-220.

Saha, P. (2006). "A Real Options Perspective to Enterprise Architecture as an Investment Activity." Journal of Enterprise Architecture 2(3): 32-52.

Saleh, Y. and M. Alshawi (2005). "An alternative model for measuring the success of IS projects: the GPIS model." Journal of Enterprise Information Management 18(1): 47-63.

Schekkerman, J. (2005). The Economic Benefits of Enterprise Architecture. New Bern, USA, Trafford.

Schmidt, J. (2005). "Valuing Enterprise Architecture." Online Features Retrieved 21.8., 2006, from http://www2.darwinmag.com/read/feature/jan05_eavalue.cfm.

Syntel. (2005). "Evaluating Your Enterprise Architecture." Applications White Paper Series Retrieved 22 August, 2006, from http://www.syntelinc.com/uploadedfiles/Syntel_EvaluateEnterArchit.pdf.

Whitman, L., K. Ramachandran and V. Ketkar (2001). A Taxonomy of a Living Model of the Enterprise. Proceedings of the 2001 Winter Simulation Conference. B. A. Peters, Smith, J.S., Medeiros, D.J., Rohrer, M.W. Arlington, Virginia, IEEE Computer Society: 848-855.

2.2 Software architecture evaluation methods

As is the case with software architecture metrics, the evaluation methods and approaches of software architecture are more established than enterprise architecture evaluation methods. Similar to the enterprise architecture domain, also here metrics and methods interrelate and this the categories include a number of same references. It should be noted that as enterprise architecture includes the software architecture domain as well, some of the methods described in these references can be used to evaluate the IT viewpoints of enterprise architecture.

AlSharif, M., W. P. Bond and T. Al-Otaiby (2004). Assessing the complexity of software architecture. Proceedings of the 42nd annual Southeast regional conference. Huntsville, Alabama, USA, ACM Press: 98-103.

Asundi, J., R. Kazman and M. Klein (2001). Using Economic Considerations to Choose Among Architecture Design Alternatives, The Software Engineering Institute, Carnegie Mellon University.

Babar, M. A. and I. Gorton (2004). Comparison of Scenario-Based Software Architecture Evaluation Methods. Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSE'04), IEEE Computer Society: 600-607.

Babar, M. A., L. Zhu and R. Jeffery (2004). A Framework for Classifying and Comparing Software Architecture Evaluation Methods. Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04), IEEE Computer Society.

Bachmann, F., L. Bass and M. Klein (2002). Illuminating the Fundamental Contributors to Software Architecture Quality, The Software Engineering Institute, Carnegie Mellon University.

Bahsoon, R. and W. Emmerich (2003). Evaluating Software Architectures: Development, Stability, and Evolution. Proceedings of ACS/IEEE International Conference on Computer Systems and Applications. Tunis, Tunisia: 47-57.

Balsamo, S., P. Inverardi and C. Mangano (1998). An approach to performance evaluation of software architectures. Proceedings of the 1st international workshop on Software and performance WOSP '98, Santa Fe, New Mexico, United States, ACM Press.

Barbacci, M., P. Clements, A. Lattanze, L. Northrop and W. Wood (2003a). Using the Architecture Tradeoff Analysis Method (ATAM) to Evaluate the Software Architecture for a Product Line of Avionics Systems: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M., R. Ellison, C. Weinstock and W. Wood (2000). Quality Attribute Workshop Participant's Handbook, Technical Report CMU/SEI-2000-SR-001, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R. (2002). SEI Architecture Analysis Techniques and When to Use Them, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., S. J. Carriere, P. H. Feiler, R. Kazman, M. H. Klein, H. F. Lipson, T. A. Longstaff and C. B. Weinstock (1997a). Steps in an Architecture Tradeoff Analysis Method: Quality Attribute Models and Analysis, Technical Report CMU/SEI-97-TR-029, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., R. Ellison, A. J. Lattanze, J. A. Stafford, C. B. Weinstock and W. G. Wood (2003b). Quality Attribute Workshops (QAWs), Third Edition, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., R. Ellison, J. A. Stafford, C. B. Weinstock and W. G. Wood (2001). Quality Attribute Workshops, The Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R., M. H. Klein and C. B. Weinstock (1997b). Principles for Evaluating the Quality Attributes of a Software Architecture, Technical Report CMU/SEI-96-TR-036, Software Engineering Institute, Carnegie Mellon University.

Barbacci, M. R. and W. G. Wood (1999). Architecture Tradeoff Analyses of C4ISR Products, The Software Engineering Institute, Carnegie Mellon University.

Bass, L., M. Klein and G. Moreno (2001). Applicability of General Scenarios to the Architecture Tradeoff Analysis Method, The Software Engineering Institute, Carnegie Mellon University.

Benarif, S., A. Ramdane-Cherif, N. Levy and F. Losavio (2004). Intelligent Tool Based-Agent for Software Architecture Evaluation. Proceedings of the Fourth International Conference on Quality Software (QSIC'04), IEEE Computer Society: 126-133.

Bengtsson, P. (1999). Design and Evaluation of Software Architecture. Department of Software Engineering and Computer Science. Karlskrona, University of Karlskrona/Ronneby.

Bengtsson, P. and J. Bosch (1998). Scenario-Based Software Architecture Reengineering. Proceedings of the 5th International Conference on Software Reuse. Victoria, BC, IEEE Computer Society: 308-317.

Bengtsson, P. and J. Bosch (1999). Architecture Level Prediction of Software Maintenance. Proceedings of the Third European Conference on Software Maintenance and Reengineering. Amsterdam, IEEE Computer Society: 139-147.

Bengtsson, P., N. Lassing, J. Bosch and H. van Vliet (2004). "Architecture-Level Modifiability Analysis (ALMA)." Journal of Systems and Software 69(1-2): 129-147.

Bergey, J., M. Barbacci and W. Wood (2000). Using Quality Attribute Workshops to Evaluate Architectural Design Approaches in a Major System Acquisition: A Case Study, Technical Note CMU/SEI-2000-TN-010, Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and M. J. Fisher (2001). Use of the Architecture Tradeoff Analysis Method (ATAM) in the Acquisition of Software-Intensive Systems, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K., M. J. Fisher and L. G. Jones (2002). Use of the Architecture Tradeoff Analysis Method (ATAM) in Source Selection of Software-Intensive Systems, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K., M. J. Fisher, L. G. Jones and R. Kazman (1999). Software Architecture Evaluation with ATAM in the DoD System Acquisition Context, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and W. G. Wood (2002). Use of Quality Attribute Workshops (QAWs) in Source Selection for a DoD System Acquisition: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Bosch, J. and P. Molin (1999). Software Architecture Design: Evaluation and Transformation. Proceedings of the IEEE Conference and Workshop on Engineering of Computer-Based Systems, ECBS '99. Nashville, TN, USA, IEEE Computer Society: 4-10.

Bot, S., C.-H. Lung and M. Farrell (1996). A Stakeholder-Centric Software Architecture Analysis Approach. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, USA, ACM Press: 152-154.

Briand, L. C., J. Carrière, R. Kazman and J. Wüst (1998a). A Comprehensive Framework for Architecture Evaluation. Technical Report ISERN-98-28. Kaiserslautern, Germany, Fraunhofer Institute For Experimental Software Engineering.

Briand, L. C., S. J. Carrière, R. Kazman and J. Wüst (1998b). COMPARE: A Comprehensive Framework for Architecture Evaluation. ECOOP'98 Workshop Reader, Lecture Notes in Computer Science (LNCS). S. Demeyer and J. Bosch, Springer-Verlag. 1543: 48-49.

Choi, H. and K. Yeom (2002). An Approach to Software Architecture Evaluation with the 4+1 View Model of Architecture. Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02), IEEE Computer Society: 286-293.

Chrissis, M. B., M. Konrad and S. Shrum (2003). CMMI: Guidelines for Process Integration and Product Improvement, Addison Wesley.

Clements, P., R. Kazman and M. Klein (2001). Evaluating Software Architectures: Methods and Case Studies. Boston, USA, Addison-Wesley.

Clements, P. C. (2000). Active Reviews for Intermediate Designs, CMU/SEI-2000-TN-009, Software Engineering Institute (SEI), Carnegie Mellon University.

Dias, M. S. and M. E. R. Vieira (2000). Software Architecture Analysis Based on Statechart Semantics. Proceedings of the Tenth International Workshop on Software Specification and Design. San Diego, CA, IEEE Computer Society: 133-137.

Dias, O. P., I. C. Teixeira and J. P. Teixeira (1999). "Metrics and Criteria for Quality Assessment of Testable Hw/Sw Systems Architectures." Journal of Electronic Testing: Theory and Applications 14: 149-158.

Dobrica, L. and E. Niemelä (2002). "A Survey on Software Architecture Analysis Methods." IEEE Transactions on Software Engineering 28(7): 638-653.

Dolan, T., J. (2001). Architecture Assessment of Information-System Families. Eindhoven, The Netherlands, Technische Universiteit Eindhoven: 222.

Dolan, T., R. Weterings and J. C. Wortmann (2000). Stakeholder-Centric Assessment of Product Family Architecture. Practical Guidelines for Information System Interoperability and Extensibility. Proceedings of the International Workshop on Software Architectures for Product Families (IW-SAPF-3), Lecture Notes in Computer Science (LNCS). F. v. d. Linden. Las Palmas de Gran Canaria, Spain, Springer-Verlag. 1951: 225-243.

Eickelmann, N. S. and D. J. Richardson (1996). An evaluation of software test environment architectures. Proceedings of the 18th international conference on Software engineering, Berlin, Germany, IEEE Computer Society.

Fabbrini, F., M. Fusani and S. Gnesi (1998). Quality Evaluation based on Architecture Analysis. Proceedings of the International Workshop on the Role of Software Architecture in Testing and Analysis (ROSATEA'98). Marsala, Italy.

Folmer, E. and J. Bosch (2004). "Architecting for Usability: a Survey." The Journal of Systems and Software 70: 61-78.

Folmer, E., J. van Gurp and J. Bosch (2003). Scenario-Based Assessment of Software Architecture Usability. Proceedings of the International Conference on Software Engineering (ICSE 2003). Portland, Oregon, USA: 61-68.

Gallagher, B. P. (2000). Using the Architecture Tradeoff Analysis Method to Evaluate a Reference Architecture: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Golden, E. J., B.E.; Bass, L. (2005). Quality vs. quantity: comparing evaluation methods in a usability-focused software architecture modification task. International Symposium on Empirical Software Engineering.

Grassi, V. M., R. (2003). "Derivation of Markov models for effectiveness analysis of adaptable software architectures for mobile computing." IEEE Transactions on Mobile Computing 2(2): 114 - 131.

Gustafsson, J., J. Paakki, L. Nenonen and A. I. Verkamo (2002). Architecture-Centric Software Evolution by Software Metrics and Design Patterns. Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02). Budapest, IEEE Computer Society: 108-115.

Hilliard, R., M. Kurland, J., S. Litvintchouk, D., T. Rice and S. Schwarm (1996). Architecture Quality Assessment, version 2.0, The MITRE Corporation.

Hilliard, R., M. J. Kurland and S. D. Litvintchouk (1997). MITRE's Architecture Quality Assessment. Proceedings of the Software Engineering & Economics Conference.

Ionita, M. T., P. America, D. K. Hammer, H. Obbink and J. J. M. Trienekens (2004). A scenario-driven approach for value, risk, and cost analysis in system architecting for innovation. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Ionita, M. T., D. K. Hammer and H. Obbink (2002). Scenario-Based Software Architecture Evaluation Methods: An Overview. Proceedings of the International Conference on Software Engineering (ICSE 2002). Orlando, Florida, USA.

Jones, L. G. and A. J. Lattanze (2001). Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R. (1996). Tool Support for Architecture Analysis and Design. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops. San Francisco, California, United States: 94-97.

Kazman, R., G. Abowd, L. Bass and P. Clements (1996). "Scenario-Based Analysis of Software Architecture." IEEE Software 13(6): 47-55.

Kazman, R., J. Asundi and M. Klein (2001). Quantifying the Costs and Benefits of Architectural Decisions. Proceedings of the 23rd International Conference on Software Engineering. Toronto, Ontario, Canada, IEE Computer Society: 297-306.

Kazman, R., M. Barbacci, M. Klein, S. J. Carrière and S. G. Woods (1999). Experience with Performing Architecture Tradeoff Analysis. Proceedings of the 21st international conference on Software engineering. Los Angeles, California, United States, IEEE Computer Society Press: 54-63.

Kazman, R., L. Bass, G. Abowd and M. Webb (1994). SAAM: A Method for Analyzing the Properties of Software Architectures. Proceedings of the 16th International Conference on Software Engineering: 81-90.

Kazman, R., L. Bass, G. Abowd and M. Webb (1998a). SAAM: A Method for Analyzing the Properties of Software Architectures. Proceedings of the 16th International Conference on Software Engineering: 81-90.

Kazman, R. and M. Klein (1998). Performing Architecture Tradeoff Analysis. Proceedings of the third international workshop on software architecture. Orlando, Florida, USA, ACM Press: 85-88.

Kazman, R., M. Klein, M. Barbacci, T. Longstaff, H. Lipson and J. Carriere (1998b). The Architecture Tradeoff Analysis Method, Technical Report CMU/SEI-98-TR-008, Software Engineering Institute, Carnegie Mellon University.

Kazman, R., M. Klein and P. Clements (2000). ATAM: Method for Architecture Evaluation, Technical Report CMU/SEI-2000-TR-004, Software Engineering Institute, Carnegie Mellon University.

Lassing, N., P. Bengtsson, H. van Vliet and J. Bosch (2002a). "Experiences with ALMA: Architecture-Level Modifiability Analysis." Journal of Systems and Software 61(1): 47-57.

Lassing, N., D. Rijsenbrij and H. van Vliet (2002b). Experience with ALMA. Proceedings ICSE 2002 Workshop on Methods and Techniques for Software Architecture Review and Assessment, IEEE Computer Society: 15-18.

Lindvall, M., R. T. Tvedt and P. Costa (2003). "An Empirically-Based Process for Software Architecture Evaluation." Empirical Software Engineering 8(1): 83-108.

Lopez, M. (2000). An Evaluation Theory Perspective of the Architecture Tradeoff Analysis Method (ATAM). Pittsburg, USA, The Software Engineering Institute, Carnegie Mellon University.

Losavio, F., L. Chirinos, N. Lévy and A. Ramdane-Cherif (2003). "Quality Characteristics for Software Architecture." Journal of Object Technology 2(2): 133-150.

Losavio, F., L. Chirinos, A. Matteo, N. Lévy and A. Ramdane-Cherif (2004). "ISO Quality Standards for Measuring Architectures." The Journal of Systems and Software 72: 209-223.

Lung, C.-H. and K. Kalaichelvan (2000). "An Approach to Quantitative Software Architecture Sensitivity Analysis." International Journal of Software Engineering and Knowledge Engineering 10(1): 97-114.

Matinlassi, M. (2004a). Comparison of Software Product Line Architecture Desing Methods: COPA, FAST, FORM, KobrA and QADA. Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE Computer Society.

Matinlassi, M. (2004b). Evaluating the portability and maintainability of software product family architecture: terminal software case study. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Matinlassi, M., E. Niemelä and L. Dobrica (2002). Quality-Driven Architecture Design and Quality Analysis Method - A Revolutionary Initiation Approach to a Product Line Architecture. VTT Publications. Espoo, Technical Research Centre of Finland, VTT: 128.

Morisio, M., I. Stamelos and A. Tsoukias (2002). A new method to evaluate software artifacts against predefined profiles. Proceedings of the 14th international conference on Software engineering and knowledge engineering (SEKE 2002). Ischia, Italy, ACM Press: 811-818.

Nord, R. L., M. R. Barbacci, P. Clements, R. Kazman, M. Klein, L. O'Brien and J. E. Tomayko (2003). Integrating the Architecture Tradeoff Analysis Method (ATAM) with the Cost Benefit Analysis Method (CBAM), The Software Engineering Institute, Carnegie Mellon University.

Paulk, M. C., B. Curtis, M. B. Chrissis and C. V. Weber (1993). Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-024-ESC-TR-93-177, SEI.

Simone, M. D. and R. Kazman (1995). Software architectural analysis: an experience report. Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research, Toronto, Ontario, Canada, IBM Press.

Stoermer, C., F. Bachmann and C. Verhoef (2003). SACAM: The Software Architecture Comparison Analysis Method, The Software Engineering Institute, Carnegie Mellon University.

Svahnberg, M. and C. Wohlin (2002). Consensus Building when Comparing Software Architectures. Proceedings of the 14th International Conference on Product Focused Software Process Improvement (PROFES 2002), Lecture Notes in Computer Science (LNCS). M. Oivo and S. Komi-Sirviö. Rovaniemi, Finland, Springer. 2559: 436-452.

Svahnberg, M., C. Wohlin, L. Lundberg and M. Mattsson (2002). A Method for Understanding Quality Attributes in Software Architecture Structures. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering. Ischia, Italy, ACM Press: 819-826.

Vigder, M., T. McClean and F. Bordeleau (2003). "Evaluating COTS Based Architectures." Lecture Notes in Computer Science 2580: 240 - 250.

Williams, L. G. and C. U. Smith (2002). PASA: A Method for the Performance Assessment of Software Architectures. Proceedings of the 3rd Workshop on Software Performance, WOSP '02. Rome, Italy: 179-189.

Woods, S. G. and M. Barbacci (1999). Architectural Evaluation of Collaborative Agent-Based Systems, The Software Engineering Institute, Carnegie Mellon University.

Wu, C. C., E. (2005). Comparison of Web service architectures based on architecture quality properties. 3rd IEEE International Conference on Industrial Informatics, INDIN '05.

3 ARCHITECTURE MANAGEMENT BACKGROUND

3.1 Enterprise architecture management

This category includes a wide range of references related to the management and governance of enterprise architecture. The topics range from enterprise architecture frameworks, governance models, and planning and development processes, methods and tools to enterprise architecture stakeholder management and benefit realization process. Moreover, related areas such as information systems management and business-IT alignment are covered. Some references of maturity models are included as well, because they have an integrated enterprise architecture management approach. A few of the references also discuss enterprise architecture on a general level: what the approach actually means, and what is included into enterprise architecture or architecture work.

Adigun, M. O. and D. P. Biyela (2003). Modelling an enterprise for re-engineering: a case study. Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology.

Al-Mashari, M. and M. Zairi (1999). "BPR implementation process: an analysis of key success and failure factors." Business Process Management Journal 5(1): 87-112.

Anaya, V. and A. Ortiz (2005). How Enterprise Architectures Can Support Integration. Proceedings of the First International Workshop on Interoperability of Heterogeneous Information Systems, IHIS '05. Bremen, Germany, ACM Press: 25-30.

Armour, F., S. Kaisler, J. Getter and D. Pippin (2003). A UML-driven enterprise architecture case study. Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03). Big Island, Hawaii, IEEE Computer Society.

Armour, F. J., S. H. Kaisler and S. Y. Liu (1999a). "A Big-Picture Look at Enterprise Architectures." IT Professional 1(1): 35-42.

Armour, F. J., S. H. Kaisler and S. Y. Liu (1999b). "Building an Enterprise Architecture Step by Step." IT Professional 1(4): 31-39.

Avison, D., J. Jones, P. Powell and D. Wilson (2004). "Using and validating the strategic alignment model." Journal of Strategic Information Systems 13(3): 223-246.

Aziz, S., T. Obitz, R. Modi and S. Sarkar. (2005). "Enterprise Architecture: A Governance Framework - Part I: Embedding Architecture into the Organization." Infosys White Paper Retrieved 22 September, 2006, from http://www.infosys.com/enterprise-architecture/.

Aziz, S., T. Obitz, R. Modi and S. Sarkar. (2006). "Enterprise Architecture: A Governance Framework - Part II: Making Enterprise Architecture Work within the Organization." Infosys White Paper Retrieved 22 August, 2006, from http://www.infosys.com/enterprise-architecture/.

Baker, D. C. and M. Janiszewski. (2005). "7 Essential Elements of EA." Enterprise Architect Retrieved 17.6.2005, from http://www.ftponline.com/ea/magazine/summer2005/features/dbaker/.

Barnett, W. P., A.; Johnson, M.; Liles, D.H. (1994). An architecture for the virtual enterprise. IEEE International Conference on Systems, Man, and Cybernetics, 'Humans, Information and Technology', 1994. San Antonio, TX.

Benson, R. J., T. Bugnitz and B. Walton (2004). From Business Strategy to IT Action: Right Decisions for a Better Bottom Line. Hoboken, USA, John Wiley & Sons.

Bernus, P. (2003). "Enterprise Models for Enterprise Architecture and ISO9000:2000." Annual Reviews in Control 27: 211-220.

Bernus, P., L. Nemes and G. Schmidt (2003). Handbook on Enterprise Architecture. Berlin, Germany, Springer-Verlag.

Boster, M., S. Liu and R. Thomas (2000). "Getting the Most from Your Enterprise Architecture." IT Professional 2(4): 43-51.

Broadbent, M. and P. Weill (1993). "Improving business and information strategy alignment: Learning from the banking industry." IBM Systems Journal 32(1): 162-179.

Brown, C. V. (2003). "Performance Metrics for IT Human Resource Alignment." Information Systems Management 20(4): 36-42.

Bruls, W. (2003). Representing Business/IT alignment in the enterprise planning cycle. Proceedings of the Landelijk Architectuur Congres 2003 (LAC 2003). Nieuwegein, the Netherlands, Nederlands Architectuur Forum.

Carbone, J. (2004). IT Architecture Toolkit. Upper Saddle River, USA, Prentice Hall.

Chalmeta, R., C. Campos and R. Grangel (2001). "References Architectures for Enterprise Integration." The Journal of Systems and Software 57: 175-191.

Chan, Y. E. (2002). "Why Haven't We Mastered Alignment? The Importance of tthe Informal Organizational Structure." MIS Quarterly Executive 1(2).

Chan, Y. E., S. L. Huff, D. W. Barclay and D. G. Copeland (1997). "Business Strategic Orientation, Information Systems Strategic Orientation, and Strategic Alignment." Information Systems Research 8(2): 125-150.

Chung, H. M. M., G. (2002). Enterprise architecture, implementation, and infrastructure management. Proceedings of the 35th Annual Hawaii International Conference on System Sciences, HICSS.

CIO Council (2000). Architecture Alignment and Assessment Guide, Chief Information Officers Council.

CIO Council (2001). The Practical Guide to Federal Enterprise Architecture, version 1.0, Chief Information Officer Council, USA.

COBIT. (2000). "Control Objectives for Information and related Technology (COBIT), 3rd Edition." from http://www.isaca.org/cobit.htm.

Curran, C. (2005). "Link IT Investments to Business Metrics." Enterprise Architect 3(1): 16-18.

de Boer, F. S. B., M.M.; Jacob, J.; Stam, A.; van der Torre, L. (2005). Enterprise Architecture Analysis with XML. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05), IEEE Computer Society.

Duffy, J. (2001). "IT/Business Alignment: Delivering Results." CIO Analyst Corner Retrieved 30 January, 2007, from http://www.cio.com/analyst/123101_idc.html.

Ekstedt, M. (2004). Enterprise Architecture for IT Management. A CIO Decision Making Perspective on the Electric Power Industry. Industrial Information and Control Systems. Stockholm, KTH, Royal Institute of Technology.

Ekstedt, M., N. Jonsson, L. Plazaola, E. S. Molina and N. Vargas (2005). An Organization-Wide Approach for Assessing Strategic Business-IT Alignment. Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET '05). Portland, USA, PICMET.

Erder, M. and P. Pureur (2003). "QFD in the Architecture Development Process." IT Professional 5(6): 44-52.

Flaxer, D. N., A.; Vergo, J. (2005). Using Component Business Modeling to Facilitate Business Enterprise Architecture and Business Services at the US Department of Defense. IEEE International Conference on e-Business Engineering, ICEBE 2005.

Fraser, P., J. Moultrie and M. Gregory (2002). The use of maturity models/grids as a tool in assessing product development capability. IEEE International Engineering Management Conference. Cambridge, IEEE.

GAO. (2003). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." from http://www.gao.gov/new.items/d03584g.pdf.

Goedvolk, H., A. v. Schijndel, V. v. Swede and R. Tolido (2000). The Design, Development and Deployment of ICT Systems in the 21st Century: Integrated Architecture Framework (IAF), Cap Gemini Ernst and Young.

Goethals, F., M. Snoeck, W. Lemahieu and J. Vandenbulcke (2006). "Managements and enterprise architecture click: The FAD(E)E framework." Information Systems Frontiers 8(2): 67-79.

Government Accountability Office (GAO). (2003). "A Framework for Assessing and Improving Enterprise Architecture Management, v. 1.1." 2004.

Grossman, I. M. and J. Sargent. (1999). "An IT Enterprise Architecture Process Model." Retrieved 17.10.2003.

Halley, M. R. B., C. (2005). Enterprise transformation to a service oriented architecture: successful patterns. Proceedings. 2005 IEEE International Conference on Web Services, ICWS 2005.

Harmon, K. (2005). The “Systems” Nature of Enterprise Architecture. IEEE International Conference on Systems, Man and Cybernetics, 2005. Big Island, Hawaii, IEEE Computer Society. 1: 78-85.

Henderson, J. and N. Venkatraman (1990). Strategic Alignment: A model For Organizational Transformation Via Information Technology. Working Paper 3223-90. Cambridge, USA, Sloan School of Management, Massachusetts Institute of Technology.

Henderson, J. and N. Venkatraman (1992). "Aligning Business and Information Technology Domains: Strategic Planning in Hospitals." Hospital and Health Services Administrative 37(1): 71-87.

Hirschheim, R. and R. Sabherwal (2001). "Detours in the Path toward Strategic Information Systems Alignment." California Management Review 44(1): 86-108.

Hirvonen, A. (2005). Enterprise Architecture Planning in Practice: The Perspectives of Information and Communication Technology Service Provider and End-user. Department of Computer Science and Information Systems. Jyväskylä, University of Jyväskylä.

Hirvonen, A. and M. Pulkkinen (2003). Evaluation of IT Architecture Solutions - How Can an ICT Consultant Tell What Is Best for You? Proceeding of the 10th European Conference on Information Technology Evaluation, Management Centre International Limited.

Hjort-Madsen, K. (2006). Enterprise Architecture Implementation and Management: A Case Study on Interoperability. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06). J. Ralph H. Sprague. Kauai, Hawaii, IEEE Computer Society.

Hoogervorst, J. (2004). "Enterprise Architecture: Enabling Integration, Agility and Change." International Journal of Cooperative Information Systems 13(3): 213-233.

Hu, Q. and C. D. Huang (2005). Aligning IT with Firm Business Strategies Using the Balance Scorecard System. Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS'05). Big Island, Hawaii, USA, IEEE Computer Society.

Hämäläinen, N. and T. Ylimäki (2004). Architecture Management in Three IT Companies - Problems and Characteristics. Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering. EASE 2004. Edinburgh: 97-109.

IAC. (2003). "Advancing Enterprise Architecture Maturity." 2003.

IAC (2005). Advancing Enterprise Architecture Maturity, version 2.0. Fairfax, USA, Industry Advisory Council, USA. 2006.

IT Governance Institute (2005). Governance of the Extended Enterprise: Bridging Business and IT Strategies. Hoboken, USA, John Wiley & Sons.

Ives, B., S. L. Järvenpää and R. O. Mason (1993). "Global business drivers: Aligning information technology to global business strategy." IBM Systems Journal 32(1): 143-161.

Johnson, P., M. Ekstedt, E. Silva and L. Plazaola (2004). Using Enterprise Architecture for CIO Decision Making: On the Importance of Theory. Proceedings of the 2nd Annual Conference on the Systems Engineering Research (CSER).

Jonkers, H., M. Lankhorst, H. ter Doest, F. Arbab, H. Bosma and R. Wieringa (2006). "Enterprise architecture: Management tool and blueprint for the organization." Information Systems Frontiers 8(2): 63-66.

Jonkers, H. v. B., R.; Arbab, F.; de Boer, F.; Bonsangue, M.; Bosma, H.; ter Doest, H.; Groenewegen, L.; Scholten, J.G.; Hoppenbrouwers, S.; Iacob, M.-E.; Janssen, W.; Lankhorst, M.; van Leeuwen, D.; Proper, E.; Stam, A.; van der Torre, L.; van Zanten, G.V.; (2003). Towards a language for coherent enterprise architecture descriptions. Seventh IEEE International Enterprise Distributed Object Computing Conference, 2003. Brisbane, Australia, IEEE Computer Society.

Kaisler, S. H., F. Armour and M. Valivullah (2005). Enterprise Architecting: Critical Problems. Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS'05). Hawaii, USA, IEEE Computer Society.

Kamogawa, T. and H. Okada (2004). Issues of e-business implementation from enterprise architecture viewpoint. 2004 International Symposium on Applications and the Internet Workshops (SAINT 2004 Workshops), IEEE Computer Society.

Kamogawa, T. and H. Okada (2005). A Framework for Enterprise Architecture Effectiveness. Proceedings of the Second International Conference on Services Systems and Services Management (ICSSSM '05). Chongqing, China, IEEE Computer Society.

Karakaxas, A. K., B.; Zografos, V. (2000). A business object oriented layered enterprise architecture. Proceedings. 11th International Workshop on Database and Expert Systems Applications, 2000. London.

Kim, J.-W., Y.-G. Kim, J.-H. Kwon, S.-H. Hong, C.-Y. Song and D.-K. Baik (2005). An enterprise architecture framework based on a common information technology domain (EAFIT) for improving interoperability among heterogeneous information systems. Proceedings of the Third ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2005). Mt. Pleasant, USA, IEEE Computer Society.

Klashorst, L. v. d. (2001). Nedcor Technology and Operations: Enterprise Architecture (the Knowledge-Base). 3rd Annual Conference on World Wide Web Applications.

Kluge, C., A. Dietzsch and M. Rosemann (2006). How to Realize Corporate Value from Enterprise Architecture. the Proceedings of the 14th European Conference on Information Systems (ECIS 2006). Göteborg, Sweden, Association for Information Systems.

Kobryn, C. (1998). Modeling enterprise software architectures using UML. Second International Enterprise Distributed Object Computing Workshop (EDOC '98). La Jolla, USA, IEEE.

Lam-Son, L. and A. Wegmann (2005). Definition of an Object-Oriented Modeling Language for Enterprise Architecture. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05), IEEE Computer Society.

Lee, S. M., D. L. Olson, S. Trimi and K. M. Rosacker (2005). "An Integrated Method to Evaluate Business Process Alternatives." Business Process Management Journal 11(2): 198-212.

Luftman, J. (2000). "Assessing Business-IT Alignment Maturity." Communications of the Association for Information Systems 4(Article 14).

Luftman, J. (2003). "Assessing IT/Business Alignment." Information Systems Management 20(4): 9-15.

Luftman, J. and T. Brier (1999). "Achieving and Sustaining Business-IT Alignment." California Management Review 42(1): 109-112.

Luftman, J. N., P. R. Lewis and S. H. Oldach (1993). "Transforming the enterprise: The alignment of business and information technology strategies." IBM Systems Journal 32(1): 198-221.

Luftman, J. N., R. Papp and T. Brier (1999). "Enablers and Inhibitors of Business-IT Alignment." Communications of the Association for Information Systems 1(11).

Maes, R. (1999). "A Generic Framework for Information Management." PrimaVera Working Paper 1999-03 Retrieved 31 January, 2007, from http://imwww.fee.uva.nl/~maestro/PDF/99-03.pdf.

Maes, R., D. Rijsenbrij, O. Truijens and H. Goedvolk. (1999). "A Generic Framework for Information Management." PrimaVera Working Paper 1999-03 Retrieved 31 January, 2007, from http://imwww.fee.uva.nl/~maestro/PDF/99-03.pdf.

Maes, R., D. Rijsenbrij, O. Truijens and H. Goedvolk. (2000). "Redefining business-IT alignment through a unified framework." PrimaVera Working Paper 2000-19 Retrieved 29 January, 2007, from http://primavera.fee.uva.nl/PDFdocs/2000-19.pdf.

McGovern, J., S. W. Ambler, M. Stevens, J. Linn, V. Sharan and E. K. Jo (2003). A Practical Guide to Enterprise Architecture. New Jersey, Prentice Hall PTR.

MEGA. (2006). "Use Enterprise Architecture as a Change Management Agent With MEGA." from http://www.mega.com/index.asp/l/en/c/solution/p/it-architecture-governance/p2/enterprise-architecture.

META Group Inc. (2000). "Architecture Capability Assessment." META Practice 4(7).

META Group Inc. (2004). "Planning the Enterprise Architecture Measurement Program." META Group Research, Enterprise Planning & Architecture Strategies, Practice Summary, from http://www.metagroup.com/us/displayArticle.do?oid=50037.

Morganwalp, J. and A. P. Sage (2003). "A System of Systems Focused Enterprise Architecture Framework and an Associated Architecture Development Process." Information Knowledge Systems Management 3(2): 87-105.

Morganwalp, J. M. and A. P. Sage (2004). "Enterprise Architecture Measures of Effectiveness." International Journal of Technology, Policy and Management 4(1): 81-94.

Mukherji, R., C. Egyhazy and M. Johnson (2002). "Architecture for a large healthcare information system." IT Professional 4(6): 19-27.

Nevelow, M. (2005). "Managing enterprise culture and politics." Journal of Enterprise Architecture 1(2): 45-51.

Niemi, E. (2006a). Architectural Work Status: Challenges and Developmental Potential - A Case Study of Three Finnish Business Enterprises. Proceedings of the 6th WSEAS International Conference on Applied Computer Science (ACS'06). Puerto de la Cruz, Tenerife, Spain, World Scientific and Engineering Society and Academy (WSEAS) Press.

Niemi, E. (2006b). Enterprise Architecture Benefits: Perceptions from Literature and Practice. Proceedings of the 7th IBIMA Conference on Internet & Information Systems in the Digital Age. Prescia, Italy, International Business Information Management Association (IBIMA).

Niemi, E. (2006c). "Enterprise Architecture Work Overview in Three Finnish Business Enterprises." WSEAS Transactions on Business and Economics 3(9): 628-635.

Niemi, E. (2007). Enterprise Architecture Stakeholders – A Holistic View. To be published in the Proceedings of the 13th Americas Conference on Information Systems (AMCIS 2007). Keystone, Colorado, USA, Association for Information Systems (AIS).

Nykänen, R. (2004). Liiketoiminta-arkkitehtuurien metamallintaminen. Department of Mathematical Information Technology. Jyväskylä, University of Jyväskylä.

OMB. (2004). "Guidelines for Enterprise Architecture Assessment Framework."

OMB (2005a). Federal Enterprise Architecture Program EA Assessment Framework 2.0, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB (2005b). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

OMB FEA Program Management Office. (2004). "Guidelines for Enterprise Architecture Assessment Framework."

OMB FEA Program Management Office (2005). OMB Enterprise Architecture Assessment Framework Version 1.5, OMB FEA Program Management Office, The Executive Office of the President, USA.

Paras, G. (2005). Enterprise architecture: Seeing the big picture. Federal Times. Springfield, USA.

Paras, G. (2006). "Building & Managing the Virtual EA Team." Architecture & Governance 2(4): 4-8.

PEER Center. (2006). "Glossary of Terms." Retrieved 9 November, 2006.

Pereira, C. M. and P. Sousa (2003). Getting into the misalignment between Business and Information Systems. Proceedings of the 10th European Conference on Information Technology Evaluation. Madrid, Spain, Academic Conferences International.

Pereira, C. M. and P. Sousa (2004). A method to define an Enterprise Architecture using the Zachman Framework. Proceedings of the 2004 ACM symposium on Applied computing. Nicosia, Cyprus, ACM Press.

Perera, D. (2005). "The good and the bad of enterprise architecture." Federal Computer Week Retrieved 12.12.2005, from http://www.fcw.com.

Perkins, A. (1998). Enterprise architecture and object-oriented development. TOOLS 26. Proceedings. Technology of Object-Oriented Languages, 1998. Santa Barbara, CA.

Plazaola, L., E. Silva, N. Vargas, J. Flores and M. Ekstedt (2006). A Metamodel for Strategic Business and IT Alignment Assesment. The Proceeding of the Fourth Annual Conference on Systems Engineering Research (CSER 2006). Los Angeles, California, USA, University of Southern California.

Pulkkinen, M. (2005). Creating a Process Model for Enterprise Architecture Management and Discrete EA Planning and Development Projects. Support for Expert Work. Department of Computer Science and Information Systems. Jyväskylä, University of Jyväskylä.

Pulkkinen, M. (2006). Systemic Management of Architectural Decisions in Enterprise Architecture Planning. Four Dimensions and Three Abstraction Levels. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06). Kauai, Hawaii, IEEE Computer Society.

Pulkkinen, M. and A. Hirvonen (2005). EA Planning, Development and Management Process for Agile Enterprise Development. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05). Hawaii, USA, IEEE Computer Society.

Rajput, V. (2004). "Strategies for Operational Risk Management." Enterprise Architect 2(3): 6-11.

Rehkopf, T. W. and N. Wybolt (2003). "Top 10 Architecture Land Mines." IT Professional 5(6): 36-43.

Reich, B. H. and I. Benbasat (1996). "Measuring the linkage between business and information technology objectives." MIS Quarterly 20(1).

Ross, J. (2004a). Generating Strategic Benefits from Enterprise Architecture. CISR Research Briefings 2004, Volume V, CISR WP No. 351, Center for Information Systems Research, Sloan School of Management.

Ross, J. (2004b). Maturity Matters: How Firms Generate Value from Enterprise Architecture. CISR Research Briefings 2004, Volume V, CISR WP No. 351, Center for Information Systems Research, Sloan School of Management.

Ross, J. W. and P. Weill (2005). Understanding the Benefits of Enterprise Architecture. CISR Research Briefings 2005, Volume V, CISR WP No. 357, Center for Information Systems Research, Sloan School of Management.

Rudawitz, D. (2003a). "Selecting an Enterprise Architecture Tool - A primer for how and why."

Rudawitz, D. (2003b). "Why Enterprise Architecture Efforts Often Fall Short." Retrieved October, 6, 2003.

Saha, P. (2004). "A Real Options Perspective to Enterprise Architecture as an Investment Activity." 2005.

Saha, P. (2006). "A Real Options Perspective to Enterprise Architecture as an Investment Activity." Journal of Enterprise Architecture 2(3): 32-52.

Saleh, Y. and M. Alshawi (2005). "An alternative model for measuring the success of IS projects: the GPIS model." Journal of Enterprise Information Management 18(1): 47-63.

Sauer, C. and L. Willcocks (2004). Strategic Alignment Revisited: Connecting Organizational Architecture and IT Infrastructure. Proceedings of the 37th Hawaii International Conference on System Sciences.

Silva, E., L. Plazaola and M. Ekstedt (2006). Strategic Business and IT Alignment: A Prioritized Theory Diagram. Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET '06). Istanbul, Turkey, PICMET.

Sledgianowski, D. and J. Luftman (2005). "IT-Business Strategic Alignment Maturity: A Case Study." Journal of Cases on Information Technology 8(2): 102 -120.

Slot, R. (2000). "Improving Business - IT Alignment Towards an Enterprise Architecture Maturity Model." Retrieved 30 January, 2007, from http://www.serc.nl/lac/LAC-2001/lac-2000/1-dynamiek/enterprise%20architecture%20maturity%20model.doc.

Sowa, J. F. and J. A. Zachman (1992). "Extending and Formalizing the Framework for Information Systems Architecture." IBM Systems Journal 31(3): 590-616.

Steen, M. W. A., D. H. Akehurst, H. W. L. ter Doest and M. M. Lankhorst (2004). Supporting viewpoint-oriented enterprise architecture. Eighth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2004), IEEE Computer Society.

Sullivan, S. (2004). "From Inception to Implementation: Delivering Business Value Through Enterprise Architecture." Rational software White Paper.

Syntel. (2005). "A Global Vision for Enterprise Architecture." Applications White Paper Series Retrieved 2 June, 2006, from http://www.syntelinc.com/uploadedfiles/Syntel_GlobalVisionEnterArchit.pdf.

Tang, A., J. Han and P. Chen (2004). A Comparative Analysis of Architecture Frameworks. Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04). Busan, Korea, IEEE Computer Society.

Tash, J. (2006). "What's the Value of EA?" Architecture & Governance magazine 2(2).

Teo, T. S. H. and J. S. K. Ang (1999). "Critical success factors in the alignment of IS plans with business plans." International Journal of Information Management 19(2): 173-185.

The Open Group. (2006). "The Open Group Architecture Framework version 8.1.1, Enterprise Edition (TOGAF 8.1.1)." Retrieved 10 September, 2006, from http://www.opengroup.org/architecture/togaf/.

Wang, G. C., A.; Wang, C.; Fung, C.; Uczekaj, S. (2004). Integrated quality of service (QoS) management in service-oriented enterprise architectures. Proceedings. Eighth IEEE International Enterprise Distributed Object Computing Conference, EDOC 2004.

Veasey, P. W. (2001). "Use of enterprise architectures in managing strategic change." Business Process Management Journal 7(5): 420-436.

Wegmann, A., P. Balabko, L.-S. Lê, G. Regev and I. Rychkova (2005). A Method and Tool for Business-IT Alignment in Enterprise Architecture. Proceedings of the 17th International Conference on Advanced Information Systems Engineering (CAiSE'05). Porto, Portugal, Springer.

Wegmann, A. P., O. (2003). MDA in enterprise architecture? The living system theory to the rescue. Proceedings. Seventh IEEE International Enterprise Distributed Object Computing Conference.

Weiss, J. W. and D. Anderson (2004). Aligning Technology and Business Strategy: Issues & Frameworks, A Field Study of 15 Companies. Proceedings of the 37th Hawaii International Conference on System Sciences (HICCS'04). Big Island, Hawaii, USA, IEEE Computer Society.

Versteeg, G. and H. Bouwman (2006). "Business architecture: A new paradigm to relate business strategy to ICT." Information Systems Frontiers 8(2): 91-102.

Whitman, L., K. Ramachandran and V. Ketkar (2001). A Taxonomy of a Living Model of the Enterprise. Proceedings of the 2001 Winter Simulation Conference. B. A. Peters, Smith, J.S., Medeiros, D.J., Rohrer, M.W. Arlington, Virginia, IEEE Computer Society: 848-855.

Whyte, M. (2005). Enterprise Architecture - The Key to Benefits Realization. DM Review White Paper. Brookfield, USA, DM Review. 2006.

Wong-Bushby, I., R. Egan and C. Isaacson (2006). A Case Study in SOA and Re-architecture at Company ABC. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06). Kauai, Hawaii, IEEE Computer Society.

Ylimäki, T. and V. Halttunen (2005). Perceptions on Architecture Management and the Skills of an Architect. Proceedings of the IBIMA 2005 Conference on Information Management in Modern Enterprise. Lisbon, Portugal.

Ylimäki, T., V. Halttunen, M. Pulkkinen and T. Lindström (2005). Methods and Tools for Enterprise Architecture - Larkki Project, Publications of the Information Technology Research Institute 16, University of Jyväskylä.

Zachman, J. A. (1987). "A Framework for Information Systems Architecture." IBM Systems Journal 26(3): 276-292.

Zarvic, N. and R. J. Wieringa (2006). An Integrated Enterprise Architecture Framework for Business-IT Alignment. Proceedings of Workshops and Doctoral Consortium of the 18th International Conference on Advanced Information Systems Engineering (CAiSE'06). Luxembourg, Namur University Press.

Zijden, S., H. Goedvolk and D. Rijsenbrij. (2000). "Architecture: Enabling Business and IT Alignment in Information System Development." Cap Gemini White Paper, from http://home.hetnet.nl/~daan.rijsenbrij/arch/publ/artarc05.doc.

3.2 Software architecture management

Similar to the previous topic, this category includes an extensive range of references on software architecture management in general. In this category, the references mostly cover software architecture planning and development frameworks and models. Several references on the concept of software or system architecture, such as architectural views, are also listed. A number of related topics such as information systems management are also included.

(2003). STRAW'03, Second International SofTware Requirements to Architectures Workshop. ICSE'03 International Conference on Software Engineering, Portland, Oregon.

Abowd, G. D. (1995). Defining reference models and software architectural styles for cooperative systems. ACM SIGOIS Bulletin, Special issue: workshop write-ups and positions papers from CSCW'94.

Ambriola, V. and A. Kmiecik (2002). Architectural Transformations. Proceedings of the 14th international conference on Software engineering and knowledge engineering. Ischia, Italy, ACM Press: 275-278.

Bachmann, F. and L. Bass (2001). Managing variability in software architectures. Proceedings of the 2001 symposium on Software reusability: putting software reuse in context SSR '01, Toronto, Ontario, Canada, ACM Press.

Bass, L., P. Clements and R. Kazman (1998). Software Architecture in Practice, Addison-Wesley.

Bass, L., P. Clements and R. Kazman (2003). Software Architecture in Practice, Addison-Wesley.

Bass, L. and R. Kazman (1999). Architecture-Based Development, The Software Engineering Institute, Carnegie Mellon University.

Bastarrica, M. C. O., S.F.; Rossel, P.O. (2004). Integrated notation for software architecture specifications. 24th International Conference of the Chilean Computer Science Society, 2004. SCCC 2004.

Bergey, J. K. and P. C. Clements (2005a). Software Architecture in DoD Acquisition: A Reference Standard for a Software Architecture Document, The Software Engineering Institute, Carnegie Mellon University.

Bergey, J. K. and P. C. Clements (2005b). Software Architecture in DoD Acquisition: An Approach and Language for a Software Development Plan, The Software Engineering Institute, Carnegie Mellon University.

Boehm, B. (1994). Software architectures: critical success factors and cost drivers. Proceedings of the 16th International Conference on Software Engineering. Sorrento, Italy, IEEE Computer Society: 365.

Boehm, B. P., D. (1999). "When models collide: lessons from software systems analysis." IT Professional 1(1): 49 - 56.

Bokhari, R. H. (2005). "The relationship between system usage and user satisfaction: a meta-analysis." The Journal of Enterprise Information Management.

Bradbury, J. S., J. R. Cordy, J. Dingel and M. Wermelinger (2004). A survey of self-management in dynamic software architecture specifications. Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems WOSS '04, Newport Beach, California, ACM Press.

Bratthall, L. and C. Wohlin (2002). "Is It Possible to Decorate Graphical Software Design and Architecture Models with Qualitative Information? - An experiment." IEEE Transactions on Software Engineering 28(12): 1181- 1193.

Chen, H.-M., R. Kazman and A. Garg (2005). "BITAM: An engineering-principled method for managing misalignments between business and IT architectures." Science of COmputer Programming 57(1): 5-29.

Dashofy, E. M. v. d. H., A.; Taylor, R.N. (2001). A highly-extensible, XML-based architecture description language. Proceedings. Working IEEE/IFIP Conference on Software Architecture, 2001., Amsterdam.

DeLone, W. and E. McLean (1992). "Information Systems Success: The Quest for the Dependent Variable." Information Systems Reseach 3(1): 60-95.

DeLone, W. and E. McLean (2003). "The DeLone and McLean Model of Information Systems Success: A Ten-Year Update." Journal of Management Information Systems 19(4): 9-30.

Drudis, A. and J. Mendonca. "The Architecture of Web Applications." Retrieved 26.5.2005, from http://www.interex.org/pubcontent/enterprise/may01/14drudis.html.

Eickelmann, N. S. and D. J. Richardson (1996). An evaluation of software test environment architectures. Proceedings of the 18th international conference on Software engineering, Berlin, Germany, IEEE Computer Society.

Fraser, P., J. Moultrie and M. Gregory (2002). The use of maturity models/grids as a tool in assessing product development capability. IEEE International Engineering Management Conference. Cambridge, IEEE.

Garlan, D. (2000). Software architecture: a roadmap. The Conference on The Future of Software Engineering, Limerick, Ireland, ACM Press.

Garland, J. and R. Anthony (2003). Large-Scale Software Architecture. A Practical Guide using UML, John Wiley & Sons, Inc.

Georgiadis, I., J. Magee and J. Kramer (2002). Self-organising software architectures for distributed systems. Proceedings of the first workshop on Self-healing systems, Charleston, South Carolina, ACM Press.

Goseva-Popstojanova, K., A. Hassan, G. A., W. Abdelmoez, D. E. M. Nassar, H. Ammar and A. Mili (2003). "Architectural-level risk analysis using UML." IEEE Transactions on Software Engineering 29(10): 946 - 960.

Guo, J., Y. Liao and B. Parvi (2006). A Collaboration-Oriented Software Architecture Modeling System - JArchiDesigner. 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems, 2006. ECBS 2006.

Gustafsson, J., J. Paakki, L. Nenonen and A. I. Verkamo (2002). Architecture-Centric Software Evolution by Software Metrics and Design Patterns. Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02). Budapest, IEEE Computer Society: 108-115.

Hai-Shan, C. (2004). Survey on the style and description of software architecture. Proceedings. The 8th International Conference on Computer Supported Cooperative Work in Design, 2004.

Hoek, A. v. d. (1999). Configurable software architecture in support of configuration management and software deployment. Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, IEEE Computer Society Press.

Hoek, A. v. d., D. Heimbigner and A. L. Wolf (1998). Versioned software architecture. Proceedings of the third international workshop on Software architecture. Orlando, Florida, United States, ACM Press: 73 - 76.

Hämäläinen, N. (2005). Quality Management Activities in Software Architecture Management (manuscript).

Hämäläinen, N., J. Markkula, T. Ylimäki and M. Sakkinen (2006). Success and Failure Factors for Software Architecture. Proceedings of the 6th International Business Information Management Association Conference (6th IBIMA). Bonn, Germany.

Hämäläinen, N. and T. Ylimäki (2004). Architecture Management in Three IT Companies - Problems and Characteristics. Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering. EASE 2004. Edinburgh: 97-109.

ICSE (2003). STRAW'03, Second International SofTware Requirements to Architectures Workshop. ICSE'03 International Conference on Software Engineering, Portland, Oregon, ACM.

IEEE (2000). IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. New York, USA, The Institute of Electrical and Electronics Engineers.

Ingram, W. and R. D. Brown (2002). Defining and implementing a scientific analysis software architecture. Conference on Object Oriented Programming Systems Languages and Applications archive
OOPSLA 2002 Practitioners Reports, Seattle, Washington, ACM Press.

Karakaxas, A. K., B.; Zografos, V. (2000). A business object oriented layered enterprise architecture. Proceedings. 11th International Workshop on Database and Expert Systems Applications, 2000. London.

Karlsson, J. and K. Ryan (1997). "A Cost-Value Approach for Prioritizing Requirements." IEEE Software 14(5): 67-74.

Kazman, R., G. Abowd, L. Bass and P. Clements (1996). "Scenario-Based Analysis of Software Architecture." IEEE Software 13(6): 47-55.

Kazman, R., M. Barbacci, M. Klein, S. J. Carrière and S. G. Woods (1999). Experience with Performing Architecture Tradeoff Analysis. Proceedings of the 21st international conference on Software engineering. Los Angeles, California, United States, IEEE Computer Society Press: 54-63.

Kazman, R. and L. Bass (2002). "Making Architecture Reviews Work in the Real World." IEEE Software 19(1): 67-73.

Kazman, R. and S. J. Carriere (1997). Playing Detective: Reconstructing Software Architecture from Available Evidence, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R., R. L. Nord and M. Klein (2003a). A Life-Cycle View of Architecture Analysis and Design Methods, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R., L. O'Brien and C. Verhoef (2003b). Architecture Reconstruction Guidelines, Third Edition, The Software Engineering Institute, Carnegie Mellon University.

Kircher, M. and P. Jain (2004). Pattern-Oriented Software Architecture: Patterns for Resource Management, Volume 3, John Wiley & Sons.

Kobryn, C. (1998). Modeling enterprise software architectures using UML. Second International Enterprise Distributed Object Computing Workshop (EDOC '98). La Jolla, USA, IEEE.

Kramer, J. M., J. (1998). "Analysing dynamic change in distributed software architectures." IEE ProceedingsSoftware 145(5): 146 - 154.

Kruchten, P. (1995). "4+1 View Model of Architecture." IEEE Software 12(6): 42-50.

Lago, P. and H. v. Vliet (2005). Explicit assumptions enrich architectural models. Proceedings of the 27th international conference on Software engineering. St. Louis, MO, USA, ACM Press.

Lassing, N., D. Rijsenbrij and H. van Vliet (2003). "How Well can we Predict Changes at Architecture Design Time?" Journal of Systems and Software 65(2): 141-153.

Lindvall, M., R. Tesoriero and P. Costa (2002). Avoiding Architectural Degeneration: An Evaluation Process for Software Architecture. Proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS'02), IEEE Computer Society: 77-86.

Magee, S. and D. Thiele (2004). "Engineering Process Standards: State of the Art and Challenges." IT Professional 6(5): 38-44.

Maranzano, J. F., S. A. Rozsypal, G. H. Zimmerman, G. W. Warnken, P. E. Wirth and D. M. Weiss (2005). "Architecture Reviews: Practice and Experience." IEEE Software 22(2): 34-43.

Matinlassi, M. (2004). Comparison of Software Product Line Architecture Desing Methods: COPA, FAST, FORM, KobrA and QADA. Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE Computer Society.

McDonell, R. B., R. (2004). Designing an open test software architecture featuring Lockheed Martin LM-STAR/spl reg/ case study. Proceedings AUTOTESTCON 2004.

McGovern, J., S. W. Ambler, M. Stevens, J. Linn, V. Sharan and E. K. Jo (2003). A Practical Guide to Enterprise Architecture. New Jersey, Prentice Hall PTR.

Medvidovic, N. and R. Taylor, N. (2000). "A Classification and Comparison Framework for Software Architecture Description Languages." IEEE Transactions on Software Engineering 26(1): 70-93.

Melone, N. P. (1990). "A Theoretical Assessment of the User-Satisfaction Construct in Information Systems Research." Management Science 36(1): 76-91.

Meszaros, G. (1995). Patterns for Decision Making in Architectural Design: Workshop Summary. Conference on Object Oriented Programming Systems Languages and Applications, Austin, Texas, United States, ACM Press.

Michiels, S., K. Verslype, W. Joosen and B. D. Decker (2005). Towards a software architecture for DRM. Proceedings of the 5th ACM workshop on Digital rights management DRM '05, Alexandria, VA, USA, ACM Press.

Niemelä, E., M. Matinlassi and A. Taulavuori (2004). "Practical Evaluation of Software Product Family Architectures." Lecture Notes in Computer Science 3154: 130-145.

Peacock, R. (2000). "Distributed architecture technologies." IT Professional 2(3): 58 - 60.

Poulin, J. S. (1996). Evolution of a software architecture for management information systems. Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops, San Francisco, California, United States, ACM Press.

Rai, A., S. S. Lang and R. B. Welker (2002). "Assessing the Validity of IS Success Models: An Empirical Test and Theoretical Analysis." Information Systems Research 13(1): 50-69.

Robbins, J. E., N. Medvidovic, D. F. Redmiles and D. S. Rosenblum (1998). Integrating Architecture Description Languages with a Standard Design Method. Proceedings of the 1998 (20th) International Conference on Software Engineering. Kyoto, IEEE Computer Society: 209-218.

Seddon, P. B. (1997). "A Respecification and Extension of the DeLone and McLean Model of IS Success." Information Systems Research 8(3): 240-253.

Shaw, M. (2001). The Coming-of-Age of Software Architecture Research. Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001: 656-664.

Shaw, M. and D. Garlan (1996). Software Architecture: Perspectives on an Emerging Discipline, Prentice-Hall.

Sherba, S. A. and K. M. Anderson (2003). A Framework for Managing Traceability Relationships between Requirements and Architectures. Second International Software Requirements to Architectures Workshop (STRAW'03). Portland, USA, IEEE Computer Society.

Smolander, K. (2002). Four Metaphors of Architecture in Software Organizations: Finding out the Meaning of Architecture in Practice. Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02). Nara, Japan, IEEE Computer Society.

Tahvildari, L., K. Kontogiannis and J. Mylopoulos (2003). "Quality-Driven Software Re-engineering." The Journal of Systems and Software 66: 225-239.

Tekinerdogan, B. (2004). ASAAM: aspectual software architecture analysis method. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), IEEE.

Uehara, S. H., H.; Kanaya, N.; Ookubo, T.; Matsutsuka, T. (1998). Enterprise model-based software architecture with server component integration. Proceedings. Second International Enterprise Distributed Object Computing Workshop, EDOC '98.

Weck, O. d. (2002). Systems Design and System Architecture - Establishing a Common Language and Set of Methods for Systems Design and Architecture. M. I. o. T. Department of Aeronautics and Astronautics.

Wei, G. G. Z. C. C. (1997). A domain-specific software architecture. IEEE International Conference on Intelligent Processing Systems, ICIPS '97. 1997. Beijing.

Wong-Bushby, I., R. Egan and C. Isaacson (2006). A Case Study in SOA and Re-architecture at Company ABC. Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS '06). Kauai, Hawaii, IEEE Computer Society.

Yacoub, S. M. A., H.H. (2002). "A methodology for architecture-level reliability risk analysis." IEEE Transactions on Software Engineering 28(6): 529 - 547.

Zarayaraz, G., P. Thambidurai, S. Kappuswami and P. Rodrigues (2003). "New Approach to SW Architecture." SIGSOFT Software Engineering Notes 28(2): 17.

Zeng, D. D. and J. L. Zhao (2002). Achieving Software Flexibility via Intelligent Workflow Techniques. Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02), IEEE Computer Society: 606-615.

4 QUALITY AND QUALITY MANAGEMENT BACKGROUND

4.1 Quality

This category lists references related to quality in general. The viewpoints include e.g. aspects of quality, quality attributes and quality management. The targets for which quality is discussed, if mentioned, are mostly related to IT or business in general.

Barbacci, M., M. H. Klein, T. A. Longstaff and C. B. Weinstock (1995). Quality Attributes, Technical Report CMU/SEI-95-TR-021, Software Engineering Institute, Carnegie Mellon University.

Basu, R. (2004). "Six-Sigma to operational excellence: role of tools and techniques." International Journal of Six-Sigma and Competitive Advantage 1(1): 44-64.

Bevan, N. (1999). "Quality in Use: Meeting User Needs for Quality." Journal of Systems and Software 49(1): 89-96.

Boehm, B. and H. In (1996). "Identifying Quality-Requirement Conflicts." IEEE Software 13(2): 25-35.

Braa, K. and L. Øgrim (1994). "Critical View of the Application of the ISO Standard for Quality Assurance." Information Systems Journal 5: 235-252.

Brykczynski, B. (1999). "A Survey of Sofware Inspection Checklists." Software Engineering Notes 24(1): 82-89.

Bøegh, J., S. Depanfilis, B. Kitchenham and A. Pasquini (1999). "A Method for Software Quality Planning, Control, and Evaluation." IEEE Software 16(2): 69-77.

Chapurlat, V., B. Kamsu-Foguem and F. Prunet (2003). "Enterprise Model Verification and Validation: an Approach." Annual Reviews in Control 27: 185-197.

Claver, E., J. J. Tarí and J. F. Molina (2003). "Critical factors and results of quality management: an empirical study." Total Quality Management 14(1): 91-118.

Claxton, J. C. and P. A. McDougall (2000) "Measuring the Quality of Models." The Data Administration Newsletter (TDAN.com) Volume, DOI:

Colombo, R. and A. Guerra (2002). The Evaluation Method for Software Product. Proceedings of the 15th International Conference on Software & Systems Engineering & their Applications (ICSSEA 2002). Paris, France.

Coronado, R. B. and J. Antony (2002). "Critical Success Factors for the Successful Implementation of Six Sigma Projects in Organisations." The TQM Magazine 14(2): 92-99.

Dale, B. G. (1994). Managing Quality, Blackwell Publishers.

Dale, B. G. (2000). Managing Quality, Blackwell Publishers.

El Emam, K. and N. Madhavji, H. (1996). "Does Organizational Maturity Improve Quality." IEEE Software 13(5): 109-110.

Florac, W. A. (1992). Software Quality Measurement: A Framework for Counting Problems and Defects, The Software Engineering Institute, Carnegie Mellon University.

Goh, T. N. (2002). "Perspectives on Statistical Quality Engineering." The TQM Magazine 11(6): 461-466.

Hong, G. Y. and T. N. Goh (2003). "Six Sigma in Software Quality." The TQM Magazine 15(6): 364-373.

Hong, G. Y. and T. N. Goh (2004). "A Comparison of Six-Sigma and GQM Approaches in Software Development." International Journal of Six-Sigma and Competitive Advantage 1(1): 65-75.

IEEE (1998). IEEE Standard for a Software Quality Metrics Methodology, IEEE Std 1061-1998.

ISO (2000a). ISO 9000:2000. Quality Management Systems - Fundamentals and Vocabulary, ISO.

ISO (2000b). ISO 9001:2000. Quality Management Systems - Requirements, ISO.

ISO (2001). ISO/IEC 9126-1:2001, Software engineering -- Product quality -- Part 1: Quality model, ISO.

ISO (2003a). ISO/IEC TR 9126-2:2003, Software engineering -- Product quality -- Part 2: External metrics, ISO.

ISO (2003b). ISO/IEC TR 9126-3:2003, Software engineering -- Product quality -- Part 3: Internal metrics, ISO.

Kartha, C. P. (2004). "A Comparison of ISO 9000:2000 quality system standards, QS9000, ISO/TS 16949 and Baldridge criteria." The TQM Magazine 16(5): 331-340.

Lecklin, O. (2002). Laatu yrityksen menestystekijänä, Gummerus.

Lillrank, P. H. (1998). Laatuajattelu: laadun filosofia, tekniikka ja johtaminen tietoyhteiskunnassa. Helsinki, Otava.

Ma, R. (1996). "Quality System an Integral Part of Total Quality Management." Computers & Industrial Engineering 31(3/4): 753-757.

Mann, R. and D. Kehoe (1995). " Factors affecting the implementation and success of TQM." International Journal of Quality & Reliability Management 12(1): 11-23.

McAdam, R. and D. Leonard (2005). "Cross-Mapping Strategic and Quality Processes: a Best Practice Analysis." International Journal of Process Management and Benchmarking 1(1): 25-44.

Motwani, J., S. Prasad and J. Tata (2005). "The Evolution of TQM: An Empirical Analysis Using the Business Process Change Framework." The TQM Magazine 17(1): 54-66.

Noda, N. and T. Kishi (1999). On Aspect-Oriented Design - An Approach to Designing Quality Attributes. Proceedings of the Sixth Asia Pacific Software Engineering Conference (APSEC '99). Takamatsu, Japan: 230-237.

Osterweil, L., L. A. Clarke, R. A. DeMillo, S. I. Feldman, B. McKeeman, E. F. Miller and J. Salasin (1996). "Strategic Directions in Software Quality." ACM Computing Surveys 28(4): 738-750.

Papalexandris, A., G. Ioannou, G. Prastacos and K. E. Soderquist (2005). "An Integrated Methodology for Putting the Balanced Scorecard into Action." European Management Journal 23(2): 214-227.

Park, R. E., W. B. Goethert and W. A. Florac (1996). Goal-Driven Software Measurement - A Guidebook, Handbook CMU/SEI-96-HB-002, Software Engineering Institute, Carnegie Mellon University.

Parnas, D., L. and M. Lawford (2003a). "The Role of Inspection in Software Quality Assurance." IEEE Transactions on Software Engineering 29(8): 674-676.

Parnas, D., L. and D. Weiss, M. (1985). Active Design Reviews: Principles and Practices. Proceedings of the 18th International Conference on Software Engineering. London, England: 132-136.

Parnas, D. L. and M. Lawford (2003b). "Inspection's Role in Software Quality Assurance." IEEE Software 20(4): 16-20.

Paulk, M. C., D. Goldenson and D. M. White (2000). The 1999 Survey of High Maturity Organizations, Software Engineering Institute, Carnegie Mellon University.

Porter, L. J. and A. J. Parker (1993). "Total quality management - the critical success factors." Total Quality Management 4(1): 13-22.

Quazi, H. A., J. Jemangin, L. W. Kit and C. L. Kian (1998). "Critical factors in quality management and guidelines for self-assessment: The case of Singapore." Total Quality Management 9(1): 35-55.

Shanks, G., E. Tansley and R. Weber (2003). "Using Ontology to Validate Conceptual Models." Communications of the ACM 46(10): 85-89.

Stylianou, A. C. and R. L. Kumar (2000). "An Integrative Framework for IS Quality Management." Communication of the ACM 43(9): 99-104.

Takku, A. (2004). Ohjelmiston laadun ja luotettavuuden estimointi luotettavuusmallien avulla. Tietojenkäsittelytieteiden laitos. Jyväskylä, Jyväskylän yliopisto: 97.

Tarí, J. J. (2005). "Components of successful total quality management." The TQM Magazine 17(2): 182-194.

Tervonen, I. and P. Kerola (1998). "Towards deeper co-understanding of software quality." Information and Software Technology 39: 995-1003.

Wang, R. Y. (1998). "A Product Perspective on Total Data Quality Management." Communications of the ACM 41(2): 58-65.

Weyuker, E. J. (1999). "Evaluation techniques for improving the quality of very large software systems in a cost-effective way." The Journal of Systems and Software 47: 997-103.

Woody, C. (2005). Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements, The Software Engineering Institute, Carnegie Mellon University.

Zeng, D. D. and J. L. Zhao (2002). Achieving Software Flexibility via Intelligent Workflow Techniques. Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02), IEEE Computer Society: 606-615.

4.2 Quality management

This category includes several same references as the previous category, but with emphasis on quality management systems, approaches, processes and methods. A few well-known approaches such as Six-Sigma, Balanced Scorecard and Total Quality Management are covered. Some references also discuss critical success factors of quality management. Quality management is mostly discussed in regard to business in general or aspects of IT.

Basu, R. (2004). "Six-Sigma to operational excellence: role of tools and techniques." International Journal of Six-Sigma and Competitive Advantage 1(1): 44-64.

Chulani, S. R., B.; Santhanam, P.; Leszkowicz, R. (2003). Metrics for managing customer view of software quality. Proceedings. Ninth International Software Metrics Symposium, 2003.

Claver, E., J. J. Tarí and J. F. Molina (2003). "Critical factors and results of quality management: an empirical study." Total Quality Management 14(1): 91-118.

Coronado, R. B. and J. Antony (2002). "Critical Success Factors for the Successful Implementation of Six Sigma Projects in Organisations." The TQM Magazine 14(2): 92-99.

Dale, B. G. (1994). Managing Quality, Blackwell Publishers.

Dale, B. G. (2000). Managing Quality, Blackwell Publishers.

Della Volpe, R. L. N., F.S.M.; Pessoa, M.S.P.; Spinola, M. (2000). The role of software process improvement into total quality management: an industrial experience. Proceedings of the 2000 IEEE Engineering Management Society.

El Emam, K. (2001). A primer on object-oriented measurement. Proceedings. Seventh International Software Metrics Symposium, METRICS 2001. London.

Habrecht, R. (1994). Quality management in an engineering firm environment. Proceedings of the 1994 IEEE International Engineering Management Conference, 'Management in Transition: Engineering a Changing World'. Dayton North, OH.

Hass, A. M. J. J., J.; Pries-Heje, J. (1998). Does ISO 9001 increase software development maturity? Proceedings. 24th Euromicro Conference, 1998. Vasteras.

ISO (2000a). ISO 9000:2000. Quality Management Systems - Fundamentals and Vocabulary, ISO.

ISO (2000b). ISO 9001:2000. Quality Management Systems - Requirements, ISO.

Kartha, C. P. (2004). "A Comparison of ISO 9000:2000 quality system standards, QS9000, ISO/TS 16949 and Baldridge criteria." The TQM Magazine 16(5): 331-340.

Kautz, K. R., F. (2001). Software quality management and software process improvement in Denmark. Proceedings of the 34th Annual Hawaii International Conference on System Sciences.

Keyes, J. (2005). Implementing the IT Balanced Scorecard - Aligning IT with Corporate Strategy. Boca Raton, USA, Ayerbach Publications.

Koch, A. S. (2001). Personal quality management with the Personal Software Process. Proceedings. 14th Conference on Software Engineering Education and Training, 2001. Charlotte, NC.

Lecklin, O. (2002). Laatu yrityksen menestystekijänä, Gummerus.

Lillrank, P. H. (1998). Laatuajattelu: laadun filosofia, tekniikka ja johtaminen tietoyhteiskunnassa. Helsinki, Otava.

Livingston, J., K. Prosise and R. J. Altizer (1995). Process Improvement Matrix: A Tool for Measuring Progress Towards Better Quality. Proceedings of 5th International Conference on Software Quality. Austin, Texas.

Lok, R. H. W., A.J. (1997). Automated tool support for an emerging international software process assessment standard. 'Emerging International Standards'. ISESS 97., Third IEEE International Software Engineering Standards Symposium and Forum. Walnut Creek, CA.

Ma, R. (1996). "Quality System an Integral Part of Total Quality Management." Computers & Industrial Engineering 31(3/4): 753-757.

Mann, R. and D. Kehoe (1995). " Factors affecting the implementation and success of TQM." International Journal of Quality & Reliability Management 12(1): 11-23.

Motwani, J., S. Prasad and J. Tata (2005). "The Evolution of TQM: An Empirical Analysis Using the Business Process Change Framework." The TQM Magazine 17(1): 54-66.

Ogasawara, H. Y., A.; Kojo, M. (1996). Experiences of software quality management using metrics through the life-cycle. Proceedings of the 18th International Conference on Software Engineering. Berlin.

Porter, L. J. and A. J. Parker (1993). "Total quality management - the critical success factors." Total Quality Management 4(1): 13-22.

Quazi, H. A., J. Jemangin, L. W. Kit and C. L. Kian (1998). "Critical factors in quality management and guidelines for self-assessment: The case of Singapore." Total Quality Management 9(1): 35-55.

Sharp, H. W., M.; Hovenden, F.; Robinson, H. (1999). The role of `culture' in successful software process improvement. Proceedings. 25th EUROMICRO Conference. Milan.

Siakas, K. V. (2002). What has culture to do with SPI? Proceedings. 28th Euromicro Conference, 2002.

Stylianou, A. C. and R. L. Kumar (2000). "An Integrative Framework for IS Quality Management." Communication of the ACM 43(9): 99-104.

Tarí, J. J. (2005). "Components of successful total quality management." The TQM Magazine 17(2): 182-194.

Thomas, S. A. H., S.F.; Barnes, D.J. (1996). Looking for the human factors in software quality management. Proceedings. International Conference Software Engineering: Education and Practice.

Walker, A. J. (1997). Quality management applied to the development of a national checklist for ISO 9001 audits for software. 'Emerging International Standards'. ISESS 97., Third IEEE International Software Engineering Standards Symposium and Forum. Walnut Creek, CA.

Walsh, J. (1994). "Software quality management: a subjective view (or `Why things still go wrong')." Journal Engineering Management 4(3): 105 - 111.

Van Grembergen, W. and R. Saull (2001). Aligning Business and Information Technology through the Balanced Scorecard at a Major Canadian Financial Group: its Status Measured with an IT BSC Maturity Model. Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS 2001). Maui, Hawaii, IEEE Computer Society.

Wang, R. Y. (1998). "A Product Perspective on Total Data Quality Management." Communications of the ACM 41(2): 58-65.

4.3 Organizational quality and excellence

This category includes references related to the quality of organizations in general. Some extensive quality management approaches are introduces, as well as references discussing the effects of quality to the success of an organization. These references could be regarded as antecedents of the enterprise architecture thinking. They could have especially had an effect to the development of enterprise architecture critical success factors and maturity models.

Badri, M. A., D. Davis and D. Davis (1995). "A study of measuring the critical factors of quality management." International Journal of Quality & Reliability Management 12(2): 36-53.

Braa, K. and L. Øgrim (1994). "Critical View of the Application of the ISO Standard for Quality Assurance." Information Systems Journal 5: 235-252.

Chapurlat, V., B. Kamsu-Foguem and F. Prunet (2003). "Enterprise Model Verification and Validation: an Approach." Annual Reviews in Control 27: 185-197.

Claver, E., J. J. Tarí and J. F. Molina (2003). "Critical factors and results of quality management: an empirical study." Total Quality Management 14(1): 91-118.

Claxton, J. C. and P. A. McDougall (2000) "Measuring the Quality of Models." The Data Administration Newsletter (TDAN.com) Volume, DOI:

El Emam, K. and N. Madhavji, H. (1996). "Does Organizational Maturity Improve Quality." IEEE Software 13(5): 109-110.

Lecklin, O. (2002). Laatu yrityksen menestystekijänä, Gummerus.

McAdam, R. and D. Leonard (2005). "Cross-Mapping Strategic and Quality Processes: a Best Practice Analysis." International Journal of Process Management and Benchmarking 1(1): 25-44.

Papalexandris, A., G. Ioannou, G. Prastacos and K. E. Soderquist (2005). "An Integrated Methodology for Putting the Balanced Scorecard into Action." European Management Journal 23(2): 214-227.

Paulk, M. C., D. Goldenson and D. M. White (2000). The 1999 Survey of High Maturity Organizations, Software Engineering Institute, Carnegie Mellon University.

Porter, L. J. and A. J. Parker (1993). "Total quality management - the critical success factors." Total Quality Management 4(1): 13-22.

Quazi, H. A., J. Jemangin, L. W. Kit and C. L. Kian (1998). "Critical factors in quality management and guidelines for self-assessment: The case of Singapore." Total Quality Management 9(1): 35-55.

Saleh, Y. and M. Alshawi (2005). "An alternative model for measuring the success of IS projects: the GPIS model." Journal of Enterprise Information Management 18(1): 47-63.

4.4 System quality, quality attributes and metrics

The references listed in this category relate to the quality attributes and metrics of different aspects of IT, such as software and information systems. Some of the references have already been listed in the software architecture categories. Mostly, these references take a more traditional measurement approach to IT than introduced in newer software architecture literature. In addition to individual metrics, some measurement approaches and models such as the DeLone and McLean information systems success model are presented in the references.

Barbacci, M., M. H. Klein, T. A. Longstaff and C. B. Weinstock (1995). Quality Attributes, Technical Report CMU/SEI-95-TR-021, Software Engineering Institute, Carnegie Mellon University.

Bass, L., P. Clements and R. Kazman (2003). Software Architecture in Practice, Addison-Wesley.

Bevan, N. (1999). "Quality in Use: Meeting User Needs for Quality." Journal of Systems and Software 49(1): 89-96.

Boehm, B. and H. In (1996). "Identifying Quality-Requirement Conflicts." IEEE Software 13(2): 25-35.

Brykczynski, B. (1999). "A Survey of Sofware Inspection Checklists." Software Engineering Notes 24(1): 82-89.

Bøegh, J., S. Depanfilis, B. Kitchenham and A. Pasquini (1999). "A Method for Software Quality Planning, Control, and Evaluation." IEEE Software 16(2): 69-77.

Colombo, R. and A. Guerra (2002). The Evaluation Method for Software Product. Proceedings of the 15th International Conference on Software & Systems Engineering & their Applications (ICSSEA 2002). Paris, France.

DeLone, W. and E. McLean (1992). "Information Systems Success: The Quest for the Dependent Variable." Information Systems Research 3(1): 60-95.

Florac, W. A. (1992). Software Quality Measurement: A Framework for Counting Problems and Defects, The Software Engineering Institute, Carnegie Mellon University.

Giaglis, G., N. Mylonopoulos and G. Doukidis (1999). "The ISSUE methodology for quantifying benefits from information systems." Logistics Information Management 12(1/2): 50-62.

Hong, G. Y. and T. N. Goh (2003). "Six Sigma in Software Quality." The TQM Magazine 15(6): 364-373.

Hong, G. Y. and T. N. Goh (2004). "A Comparison of Six-Sigma and GQM Approaches in Software Development." International Journal of Six-Sigma and Competitive Advantage 1(1): 65-75.

IEEE (1998). IEEE Standard for a Software Quality Metrics Methodology, IEEE Std 1061-1998.

ISO (2001). ISO/IEC 9126-1:2001, Software engineering -- Product quality -- Part 1: Quality model, ISO.

ISO (2003a). ISO/IEC TR 9126-2:2003, Software engineering -- Product quality -- Part 2: External metrics, ISO.

ISO (2003b). ISO/IEC TR 9126-3:2003, Software engineering -- Product quality -- Part 3: Internal metrics, ISO.

Ivkovic, I. K., K. (2006). A Framework for Software Architecture Refactoring using Model Transformations and Semantic Annotations. Proceedings of the 10th European Conference on Software Maintenance and Reengineering, CSMR 2006.

Jones, L. G. and A. J. Lattanze (2001). Using the Architecture Tradeoff Analysis Method to Evaluate a Wargame Simulation System: A Case Study, The Software Engineering Institute, Carnegie Mellon University.

Kazman, R., J. Asundi and M. Klein (2001). Quantifying the Costs and Benefits of Architectural Decisions. Proceedings of the 23rd International Conference on Software Engineering. Toronto, Ontario, Canada, IEE Computer Society: 297-306.

Lassing, N., D. Rijsenbrij and H. van Vliet (1999). Towards a Broader View on Software Architecture Analysis of Flexibility. Proceedings of the Sixth Asia Pacific Software Engineering Conference (APSEC '99). Takamatsu, IEEE Computer Society: 238-245.

Lee, G. and W. Xia (2005). "The ability of information systems development project teams to respond to business and technology changes: a study of flexibility measures." European Journal of Information Systems 14(1): 75-92.

Noda, N. and T. Kishi (1999). On Aspect-Oriented Design - An Approach to Designing Quality Attributes. Proceedings of the Sixth Asia Pacific Software Engineering Conference (APSEC '99). Takamatsu, Japan: 230-237.

Osterweil, L., L. A. Clarke, R. A. DeMillo, S. I. Feldman, B. McKeeman, E. F. Miller and J. Salasin (1996). "Strategic Directions in Software Quality." ACM Computing Surveys 28(4): 738-750.

Palanisamy, R. (2005). "Strategic information systems planning model for building flexibility and success." Industrial Management & Data Systems 105(1): 63-81.

Park, R. E., W. B. Goethert and W. A. Florac (1996). Goal-Driven Software Measurement - A Guidebook, Handbook CMU/SEI-96-HB-002, Software Engineering Institute, Carnegie Mellon University.

Parnas, D., L. and M. Lawford (2003a). "The Role of Inspection in Software Quality Assurance." IEEE Transactions on Software Engineering 29(8): 674-676.

Parnas, D., L. and D. Weiss, M. (1985). Active Design Reviews: Principles and Practices. Proceedings of the 18th International Conference on Software Engineering. London, England: 132-136.

Parnas, D. L. and M. Lawford (2003b). "Inspection's Role in Software Quality Assurance." IEEE Software 20(4): 16-20.

Takku, A. (2004). Ohjelmiston laadun ja luotettavuuden estimointi luotettavuusmallien avulla. Tietojenkäsittelytieteiden laitos. Jyväskylä, Jyväskylän yliopisto: 97.

Tervonen, I. and P. Kerola (1998). "Towards deeper co-understanding of software quality." Information and Software Technology 39: 995-1003.

Weyuker, E. J. (1999). "Evaluation techniques for improving the quality of very large software systems in a cost-effective way." The Journal of Systems and Software 47: 997-103.

Woody, C. (2005). Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements, The Software Engineering Institute, Carnegie Mellon University.

Yakimovich, D., J. M. Bieman and V. R. Basili (1999). Software architecture classification for estimating the cost of COTS integration. Proceedings of the 21st international conference on Software engineering, Los Angeles, California, United States, IEEE Computer Society Press.

Zeng, D. D. and J. L. Zhao (2002). Achieving Software Flexibility via Intelligent Workflow Techniques. Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS'02), IEEE Computer Society: 606-615.