Anette Lavu

Designing APIs in industrial manufacturing context: a case study

Master’s thesis of mathematical information technology

August 23, 2019

Author: Anette Lavu

Contact information: anbeevla@student.jyu.fi

Supervisors: Pekka Abrahamsson, Sami Frisk

Title: Designing APIs in industrial manufacturing context: a case study

Työn nimi: APIen suunnittelu teollisessa kontekstissa: tapaustutkimus

Project: Master's thesis

Study line: Mathematical Information Technology

Page count: 112 + 9

Abstract: The purpose of this study is to reveal how application programming interfaces can be found and which ones should be implemented in manufacturing companies that don't have former experience from APIs. This study also discusses how to design, implement, manage and evaluate APIs. In the empirical part of this study it is explored which APIs should be implemented based on the prioritization.

Keywords: API Design, Exploring API ideas, Prioritizing

Suomenkielinen tiivistelmä: Tämän tutkimuksen tarkoituksena on selvittää, kuinka teolliset tuotantoyritykset voivat selvittää miten API rajapintoja voidaan löytää liiketoimintavaatimuksien perusteella ja mitä API rajapintoja näiden perusteella tulisi implementoida. Tässä tutkimuksessa tutkitaan, miten API rajapintoja suunnitellaan, implementoidaan, ylläpidetään ja arvioidaan. Tämän tutkimuksen empiirisessä osassa selvitetään mitä API rajapintoja tulisi implementoida priorisoinnin perusteella.

Avainsanat: APIen suunnittelu, API ideointi, priorisointi

Glossary

API	Application Programming Interface
B2B	Business to Business
B2C	Business to Customer
CRUD	Create, Read, Update and Delete
DevOps	Development and Operations
DCOM	Distributed Component Object Model
EULA	End-User Licence Agreement
IaaS	Infrastructure as a Service
IoT	Internet of Things
NPS	Net Promoter Score
PaaS	Platform as a Service
PoC	Proof of Concept
SaaS	Software as a Service
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SLA	Service Level Agreement
SSO	Single-Sign-On
TFFHW	Time For First Hello World
TOS	Terms of Service
UCD	User-Centered Design

Contents

A Tables

B Figures

1 Introduction

An API (Application Programming Interface) specifies how software components should interact by set of procedures, functions, rules and protocols (Ghute & Raghuwanshi, 2016). APIs can change the whole business model of an organization because they enable not only data, but also more sophisticated information processing functionalities (Wulf & Blohm, 2017). API can be a digital service product that you can sell or a way to save money, time or improve security, even though sometimes it's just a compulsory part of customer experience. Often APIs are mixed with integrations, but in fact between two APIs an integration tool is needed to enable them to communicate with each other. API is an interface, not an integration. (Moilanen, Niinioja, Seppänen & Honkanen, 2018) API-trends have developed rapidly, which has led to a point where companies might not know which direction to take with API development. Should companies follow the competition in the market and just do whatever is thriving at the moment or take a deep breath and investigate what they actually need to accomplish to create value to customers? Platform economy is a new phenomenon which have led former production and service centric organizations to think and develop new ways to operate since communication is the new key initiative in digital world (McPhee, Dedehayir & Seppänen, 2017). The business model defines what kind of combination of resources and skills are needed to create an organization's value, and how it is shared.

APIs enable business to business collaboration in real-time by creating new experiences via mobile, social and IoT interactions. By 2020 at least 25% of B2B interactions will not use legacy approaches, but APIs instead. (O'Neill & Golluscio, 2017) Benzell, Lagarda & Van Alystane (2017) found that API adoption is statistically related to an increase in net income by earning 3 million more per year than non-adopters, and overall 3 percent increase in net profits while turnover increased a staggering 13,5%. While this sounds compelling, it can't be ignored that APIs are an investment that needs continuous delivery services and maintenance - a lifecycle management from the moment it is planned, until it's no longer existing (SOA Software, 2012).

In this thesis I examine how APIs should be designed in B2B field based on business requirements in the context of case company and discuss what kind of components companies need to add to gain value from APIs. Because there is no former study examining the content of API design guidelines, (Murphy, Alliyu, Macvean, Kery & Myers, 2017) I have gathered evidence widely across the B2B and B2C API studies and found out that multiple objectives have not been studied thoroughly. It might be because the standards and methods are still developing in API scene. Specifically, Finland has been less actively opening platform resources compared to other international benchmark companies which might decrease overall development of platform economy in Finland. Hence understanding the differences between platform economy and traditional business models should be expanded to keep up with the international companies. (Still, Seppänen, Seppälä, Suominen, Valkokari & Korhonen, 2017)

APIs can be used in various contexts in businesses, hence it still might be an unexplored territory to big manufacturing companies, how to best exploit them to customers, or should they at all. APIs are products you can use as building blocks for applications to develop faster and integrate existing systems easier, which is why it's important that APIs go through an appropriate lifecycle (Ravichandran, Taylor & Waterhouse, 2016). Lifecycle should be guided by API strategy that defines expected business outcomes, partners and metrics (Holley et al., 2014). Whereas the interface that defines how to access services programmatically should have separate lifecycles from the service implementation to gain flexibility. (Dayley & Oliffe, 2017). It's also important to understand risks and how to mitigate them in cloud environment, since being aware of them will help to achieve organizations objectives (Farrag & Nasr, 2017).

Multiple manufacturing companies lack an understanding of shaping up digitalization capabilities. When companies are offering advanced product-services they should investigate how they can leverage these capabilities to co-create value with customers. (Lenka, Parida, Sjödin & Wincent, 2016) But what APIs should a company start to implement or is value assured which ever API they implement? Where do these API needs come from? API should only exist to serve a business purpose and be driven by the business requirements derived from customers (SOA Software, 2012). Hence, considering the exploratory nature of this research, an iterative approach was constructed by arranging a sample interview to qualify designed theoretical framework model for interview questions. Designing APIs starts by identifying customer and business needs, forming business requirements based on them and finally constructing a complete API idea.

1.1 Motivation

Legacy approaches shall continue even after API adoption as a majority of B2B interactions, but there are multiple advantages of APIs compared to traditional technologies in the table below (O'Neil & Golluscio, 2017). Adopting APIs will send a signal to investors giving a probable cause to see a company as adaptive and enabling new opportunities in the future. Ultimately 12.7% percent of increase in market capitalization and a strong relationship with net income to B2B API calls was founded by Benzell et al., (2017). It seems only a matter of time when APIs are the "must have" influencer when calculating market values and potential investors and customer weight whether to join a particular company or not. The effect APIs have, cannot be overlooked anymore by stating, "it's just a hype term".

1.2 Reaech questions and objectives

This study reveals how to design and develop B2B APIs in big manufacturing companies. The first research question is *how to design and develop a B2B API?* It should be explained which factors should be considered and what kind of process is needed to successfully implement APIs in B2B context.

The second research question is *what APIs offers the most value to the case company?* This question gives guidelines to form a roadmap about which APIs case company should implement in the future. Because this information is case company sensitive it's based on interview results and value evaluation is done by prioritization.

Third question is *how to find valuable APIs*? There was no former study about how to can API ideas be formed if the business objectives and strategy are lacking. Therefore, based on the literature review and the empirical research, I should construct a method where API needs will arise. Later, other companies could follow the method for revealing API ideas as well.

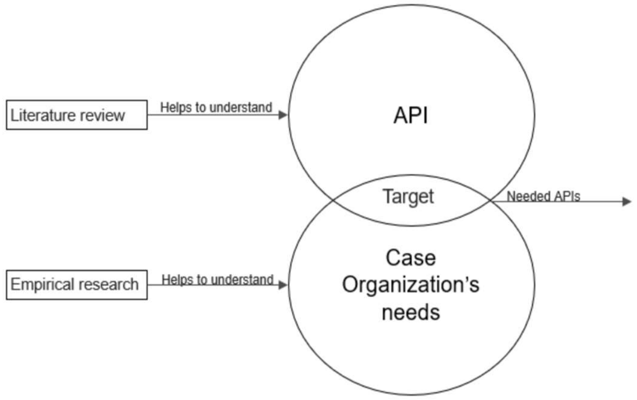


Figure 1. Research process and focus

1.3 Thesis structure

The first chapter introduces the subject and describes the research questions and methods used. In the second chapter, related work and the theoretical evidence to first research question is presented thoroughly. Theoretical framework developed based on the literature is shown in chapter three. Research design, data collection and analysis with a case company description are presented in chapter four. Finally, in the fifth chapter, empirical research results are listed and summarized, followed by discussion about theoretical and practical implications in chapter six. Conclusions are drawn in chapter seven, including answers to research questions and discussion about limitations and future research.

2 Related work

In the second section the theoretical evidence is presented to answer the research question "how to design APIs?" based on the former literature. Evidence is presented from defining the strategy for APIs to measuring and evaluating them. The whole process of an API's lifecycle is taken into notice. It is also discussed how to measure, evaluate and support APIs continuously.

API (Application Programming Interface) nowadays means multiple things from integration to atomic services combining endpoints and data with value added functionalities (Fletcher, 2018). Fielding presented RESTful web services in 2000 enabling parallel interactions to be processed and improving scalability (Fielding, 2000), which has been the first theory taking a notice into web services architecture, being the theory that enables APIs to exist in the form we know them (Moilanen et al., 2018). Digitalization, environmental and sustainability challenges are part of organization's environment nowadays, which is why it is vital to focus what is happening in the external environment. (Kangas, Heikkinen, Lönqvist, Laihonen & Bethwaite, 2019) Platform economy being the new phenomenon which has led former production and service centric organizations to think and develop new ways to operate since communication is the new key initiative in digital world (McPhee et al., 2017). To leverage business relationships in distributed environment APIs are used to collaborate in B2B field effectively allowing partners to utilize existing business relationships. (Boyd, 2015)

Spreading of SaaS (software as a service) and other cloud services has caused internal services to transform away from internal networks (Benzell et al., 2017). These pressures might easily drive companies to make hasty decisions and overlook the actual need without a clear strategy for APIs. There are numerous pitfalls that might affect API consumer negatively. Building API from a provider's point of view and neglecting consumer centric approach will end up confusing the consumers. All consumers should be treated equally, meaning that if customization is needed, a completely new API is much more efficient to make. Measuring API should concentrate on consumption and not production as much. (Dayley & Oliffe, 2017) Resource models of API should not reflect organizations internal canonical model, but instead should be carefully thought from consumer's perspective (Matheny, 2017).

The API provider is the company that creates APIs and makes them available for others, whereas API consumers are using APIs to create applications for users who are the definitive beneficiaries of the applications that use APIs. Business customers are usually organizations that make the decision to commit to using the API provider's API in their own organization (Barnes et al., 2018). In this study, the company acts as an API provider seeking business customers to commit, whereas the customer refers to another company and not to an individual customer or developer (SOA Software, 2012). APIs can be beneficial in multiple scenarios, in some cases being the actual product and other cases being a part of bigger solution, depending on the business needs. Dayley (2018) separated APIs to outer APIs, inner APIs, API and event mediation and multigrained services that can build fit-to-purpose Apps. Outer APIs are supporting the needs of apps and integrations providing consistent access for all apps and integrations, whereas inner APIs expose a single functionality. API and event mediation layer are abstraction layers acting between inner and outer APIs providing consistent access management, traffic flow and monitoring capabilities.

The decision to create an API facing customers is always a strategic decision which cannot be made hesitantly. There needs to be a plan with contracts when important data and processes are accessed (Moilanen et al., 2018). For example, a Japanese health technology company collects data from e-health and medical devices selling aggregated datasets to partners. The business benefit for using partner API for this company is data monetization: they found a way to create business streams using data that they were already collecting (Boyd, 2014). API needs to be managed like a technology product, not a transient IT project, meaning that it's vital to implement a full life cycle management, with versioning and roadmaps. API design, API lifecycle management, and support need disciplined commitment to provide good quality of adoption and satisfaction for customers. (Dayley & Oliffe, 2017) API strategy is needed to help project leaders and managers to design valuable and meaningful APIs (Ravichandran et al., 2016). API strategy shall also give insight how to measure and evaluate the impact of APIs. There are many ways you can measure the benefits of API depending on the target point of view, for example business or security (Moilanen et al., 2018). Designing a B2B API strategy is a major step towards creating more value by increasing business efficiency and enabling desired features for customers via API. (Ravichandran et al., 2016)

2.1 API Strategy

Embracing the opportunities APIs enable, organizations need a strategy, not just a set of tools. Strategy will help engender an understanding about the bigger picture where APIs are a digital enabler and not just a technology. (Ravichandran et al., 2016) Like Vijayakumar (2018) states, “any business decision involving planning, organization or governance of an API is a strategy”. API strategy is driven by organizations business strategy and needs. Strategy consists of establishing a clear vision with business objectives and building business model around API vision with detailed outlining of: costs, resources, efficiencies, value, revenue, innovation and operational process. (Ravichandran et al., 2016) Oliffe (2017) structured the API strategy into a practical guidance framework with five different categories to help the management; deploy and operate, enable developers, managing the API lifecycle, communicating securely, reliably and flexibly and measuring and improving business value, where all the management solutions need to fit with application infrastructure. Both methods seem to cover similar areas of strategical decisions, but the perspective differ. Ravichandran et al., (2016) takes more business-driven view whereas Oliffe (2017) reveals more practical approach. These two approaches are presented below in the table 1. to ease the comparison.

<table>
<tr><th colspan="2">API Strategy Business objectives</th></tr>
<tr><td>Ravichandran et al., (2016)</td><td>Oliffe (2017)</td></tr>
<tr><td>Costs, resources and efficiencies; systems, relationships and activities</td><td>Deploy and operate; management solutions must fit with application infrastructure</td></tr>
</table>

Value, revenue and innovation; customers, markets and channels of the target program, how technical innovation enable generating new revenue	Enable developers; help developers publish, support, version and retire APIs
	Manage API lifecycle; provide capabilities to publish, support, version and retire APIs
	Communicate securely, reliably and flexibly; support secure and scalable consumption
Operational processes; tools and approaches needed to control, measure, optimize and deploy multiple APIs through lifecycle	Measure and improve business value; support monitoring, analytics and monetization

Table 1. API Strategy Business Objectives Comparison

It needs to be made clear whether the desired strategy is to provide internal, external, public or combination of APIs. The decision should be based on the business need, for example a partner APIs can be made when business relationships needs management. (Boyd, 2014) Sometimes it might be beneficial to start from the internal APIs since the customer might lose trust if the API is misfunctioning. Specifically, when business is not willing to open the core business transformations to external parties and testing is needed to achieve desired efficiencies and productivity improvements. (Boyd, 2015) Goals during API planning should include: first that the business purpose for the API should be determined, and secondly that you should understand the cost/benefit outcomes for the business and users. After these are covered, it is needed to agree about the priority and delivery schedule while structuring business to support and manage the API. (SOA Software, 2012) Different API types are presented further in appendix D.

Business stakeholders should propose a vision, and technical stakeholders work towards achieving the business objectives (Vijayakumar, 2018). Boyd (2015) states that managing

API strategy might be costly which should be considered. Defining API business models requires that outcomes, development, and value strategy are planned before API creation and consumption. Transforming business assets is needed for value generation through API development, metering and analytics. There are four different maturity levels: API discovery and experimentation, platform selection and targeted expansion, re-imagining core businesses and business as a service built on API ecosystem. (Holley et al., 2014) Depending on the API Adoption model - the level of maturity presented in figure 2, the strategy for adopting APIs should differ. (Boyd, 2015)

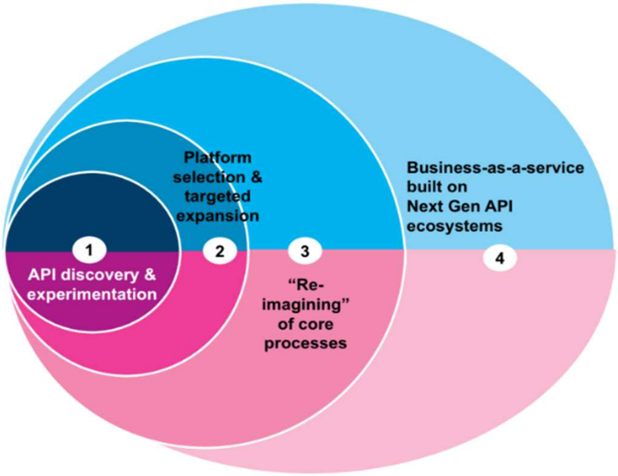

Platform selection & targeted expansion
Business-as-a-service built on Next Gen API ecosystems
1
2
3
4
API discovery & experimentation
"Re-imagining" of core processes

Figure 2. API Adoption model (Holley et al., 2014)

2.2 API Lifecycle Management

Weather talking about APIs or other products and services, there are few options to choose from, when talking about lifecycle management. In this chapter I will present few most suitable methodologies for API management and compare them briefly. Service-oriented design and development methodology presented in figure 3. is an iterative incremental process which constructs from eight phases including planning, analysis and design (A&D), construction and testing, provisioning, deployment, execution and monitoring. The focus is to deliver continuous invention, discovery and implementation forcing predictability taking technical and business concerns into account, which makes it suitable for web services. (Papazoglou & Heuvel, 2006)

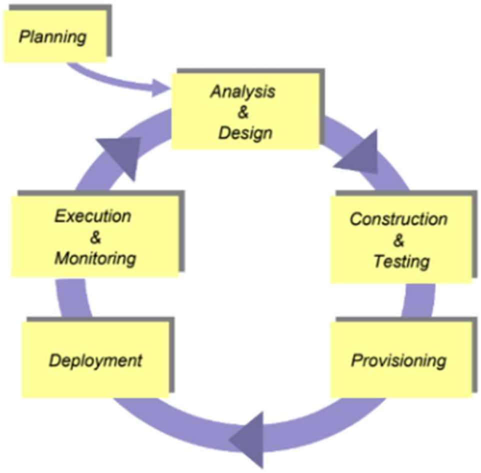


Figure 3. Service-oriented design and development methodology (Papazoglou & Heuvel, 2006)

According to Ries (2011) APIs should be developed by lean startup – a method where prioritization and adaptation is based on the value created to customers. This method consists of building, measuring, and learning– a development cycle which directs the process of API development by continuous delivery, analytics and feedback. DevOps (Development and Operations) presented in figure 4. enables lean startup- method by combining culture and a set of technologies and tools used for automation. But if organization lack DevOps culture, multiple challenges and struggles will follow because of the inability for high-velocity product development. (Familiar, 2015)

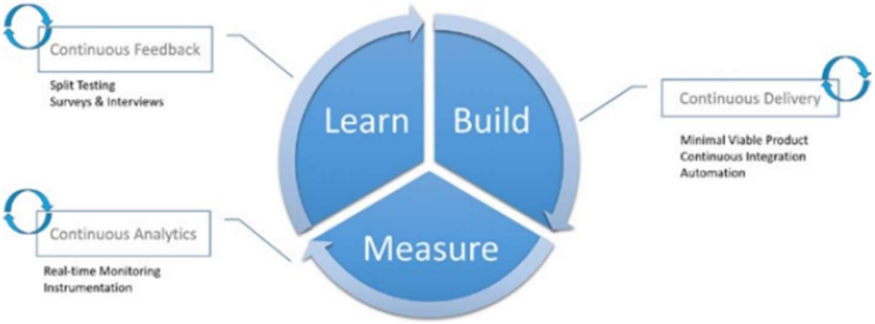


Figure 4. Lean engineering's Build-Measure-Learn Cycle (Familiar, 2015)

API is a first-class product, which means that there needs to be an API Product Manager leading life cycle management (SOA Software, 2012). API lifecycle management, or in other words API product management, includes designing, creating and running, securing, managing and optimizing the API (Ravichandran et al., 2016; SOA Software, 2012). The designing process starts by defining API requirements and specifications, identifying how API will be used and by whom, as well as leveraging existing resources as much as possible. But because the current standards for B2B APIs are lacking, they need to be built based on B2B requirements (O'Neill & Golluscio, 2017). Building face includes building client and technical environment where non-functional requirements, like security and logging should be added. When all this is done, the API is ready to be run, optimized and supported. (SOA Software, 2012) These phases are constructed in figure 5 as a simplified lifecycle management process.

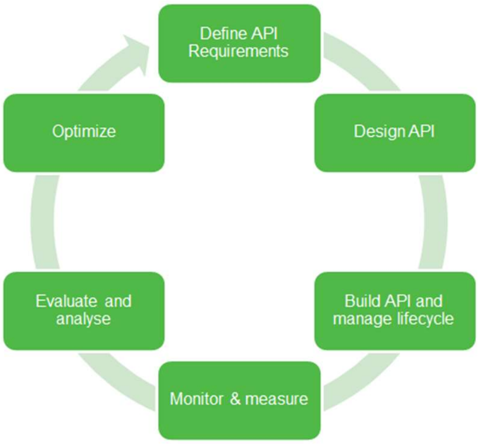


Figure 5. Process for API Lifecycle Management

API requirements are based on business needs and requirements, as well as users' needs and limitations. Business need is the reason why a new system, platform, or API is needed (Alnabhan, Haboush, Al-Badareen, Al-nawayseh & El-Zaghmouri, 2014). When APIs business needs are defined, the circumstances around the API should be known. There are two activities in requirements management: gathering requirement analysis and specifications and change management for requirements where activity checklist can be used to obtain customer sign-off (Mohapatra, 2015). All options for implementation should be considered weather API is the right approach at all, and what is the urgency of delivery.

(SOA Software, 2012) Therefore an evaluation process is needed to capture whether the proposed reasons are enough to build a new API (Alnabhan et al., 2014). Enabling efficient data sharing and reuse are the benefits of APIs because of the laborious data exploration done manually (Ferreira et al., 2018). The characteristics of API audience should also be understood thoroughly; needs, mental models, experiences and skills should be concluded into a scenario. Basic need of a customer is usually a simple definition task, but in order to understand mental models' designers need to see through customers eyes. After scenarios answer who, what, when and why, the mapping of third-party developer persona is ready to be extracted from requirements as presented in the figure 6. (Dayley & Oliffe, 2017) Even though it is risky, if not done correctly. Based on Billestrup et al. (2016) study all the participants in their studies were using personas differently and using personas incorrectly could have significant negative impact towards the product under development.

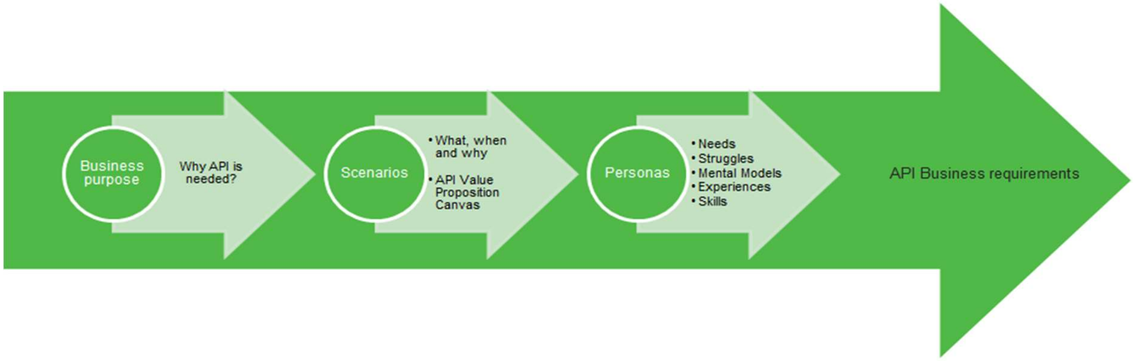


Figure 6. Process of Defining API Business Requirements

API requirements evolve continuously based on contextual factors like the environment, technology, organization, decisions made, individuals, and groups. These factors shape as well as the enterprise level software as requirements. (Scheider, Wollersheim, Krcmr & Sunyaev, 2018). The first step towards adopting an API strategy is API discovery and experimentation which includes, for example, customer facing mobile applications with limited integrations to core systems. (Holley et al., 2014) Implementing new services is inevitable when you desire to make your data and functionality as part of web APIs (Oliffe, 2018). Like mentioned earlier, APIs exist to serve a business purpose, which could be, for example, providing access to online catalogs and inventory, exposing core functions to external users, providing self-service options, or expanding the mobile device usage user base. (SOA Software, 2012) Thus Wulf & Blohm (2017) states that 90% of the freely offered data is offered as an alternative channel to a website aiming for multi-channel access, whereas 60% of the APIs in total do not only provide data but more sophisticated functionalities.

2.2.1 Business need

APIs are driven by a need (Ofoeda & Boateng, 2018). Business needs are the reason behind why a new API is needed to achieve certain or multiple goals. (Alnabhan et al., 2014) Companies might want more traffic on their website or portal, provide self-service or online access to online catalogs and inventory as well as exposing core functions. Sometimes companies already have wanted capabilities, but they want to offer them as a service, expand the use of mobile devices or build a community around the brand. (SOA Software, 2012) Similar evidence was founded by Holley et al. (2014) where growing customer base by attracting them, driving innovation, improving time-to-value and time-to-market as well as proving integrations were the main reasons why companies should implement APIs. Furthermore, some sophistication and evolving has happened within a few years, since Dayley and Oliffe (2017) categorizes business needs as: enable web and mobile interactions, integrate internal applications, interface with microservice, publish data, create cloud and SaaS integrations, enable IoT interactions, engage customers and extend business.

API business requirements are based on the potential APIs users need to solve a business problem. Without thorough understanding of customers' needs, it's probable that rework and extra cost becomes part of API lifecycle management, and that is not the goal (Mohapatra, 2015). Furthermore, sometimes it might be plausible to dive into the business problem if the business need itself is not clear. It needs to be broken down and thought carefully about what kind of value API users can gain by using this API in different phases of their processes and what kind of problems or struggles can they face (Moilanen et al., 2018). There might be some difficulties deciding the time frame between current and future needs, ensuring the participation of key stakeholders, and handling conflicting customers' needs (Mohapatra, 2015). All parties in a value chain should be identified by application leaders to enable successful API business models and understand their role in API success regarding customer profiling, positioning, monetization and support. (Barnes et al., 2018) Value chains to be recognized for APIs are presented below in the figure 7 showing the links between partner, individual developer, business customer and the final (end) user.

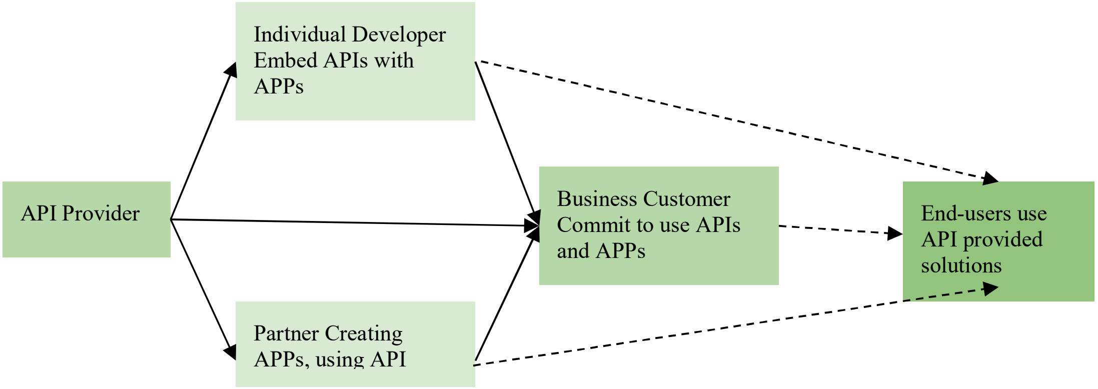

Individual Developer Embed APIs with APPs
API Provider
Business Customer Commit to use APIs and APPs
End-users use API provided solutions
Partner Creating APPs, using API

Figure 7. Value Chains for API-Enabled Solutions (Barnes et al., 2018)

2.2.2 Customer's need

APIs should be designed in a consumer-driven way when developers are the focus and not the programs. It needs to be considered that developers have specific needs, skills and preferences when developing their own services based on offered APIs. Developers should be considered by applying personas, scenarios and consumer-driven contracts, because it needs to be understood who will be using the API and how to ensure their quick and effective way of use (Dayley & Oliffe, 2017). Drawing sets of personas and defining targeted developer types together with information about who and what department they work for, why customers are developing an app, programming skills and protocol preferences, will give you complete idea about the user's point of view (CA Technologies, 2015). By involving actual users in design, Hjalmarsson, Gustafsson and Cronholm (2015) introduced an actual end user involvement in UCD (User-Centered Design) based on Abras et al.,

2014. In UCD method data is collected in the beginning of the design process with multiple techniques like interviews, questionnaires, focus groups, on-site observations, role playing, walkthroughs and testing. The rest of the design process consists of validating the requirements and evaluating alternative design and prototypes. The last phases of the design process are more or less collecting data and measuring how well are the user satisfied. The whole design process includes the end users involvement to be active and ongoing. Design phases with users involvement described in more detail below in table 2.

Design State	Purpose of end user involvement	Example technique
Beginning of the design process	Collect data related to needs and expectations, evaluation of design alternatives, prototypes and e-service	Background interviews and questionnaires
Early in design process	Collect data related sequence of work to be performed by e-service	Sequence of work interviews and questionnaires
Early in design process	Wide range of stakeholders to discuss issues and requirements	Focus group
Early in design process	Collect information about final e-services operating environment	On-site observation
Early and mid-design process	Evaluation of alternative designs and gaining more information about the users' needs and expectations, prototype evaluation	Role playing, walkthroughs and simulations
Final stage of design	Collect quantities data related to measurable usability criteria	Usability testing
Final stage of design	Collect qualitative data related to user satisfaction	Interviews and questionnaires

Table 2. Part of end user involvement in UCD (Hjalmarsson et al, 2015)

Identifying customers' needs and experiences can be accomplished by interviewing customers or actively trying to put oneself in the customers position (Moilanen et al., 2018). To create ideal customer profiles, companies need to be able to capture the key characteristics of the organizations that they believe to be their customers, including information about why this customer would be a good fit (Barnes et al., 2018). There might be some difficulties and challenges in articulation, communication and cognitive limitations. For example, a user may not be able to articulate their needs, or they are misunderstood, incompatible style of interaction (details versus abstractons), too simplified problem description, fear of automation causing retrenchment, and difficulty of scaling (Mohapatra, 2015). More articulation, communication and cognitive limitations were presented by Mohapatra (2015) and they presented in appendix G.

Adopting consumer-centric approaches with scenarios and personas to API design helps to understand and engage API consumers. To glean information about personas surveys, ethnographies and interviews or focus groups, should be used to gain data about demographics, contextualization, behaviour and attitudes. Technology is always used in a culture, a context that varies. Technology might be perceived in different ways depending on the culture which makes is important to understand cultural, technological, global and local context when developing personas. (Getto, 2014). Billestrup, Stage, Bruun, Nielsen and Nielsen's (2016) studied creating and using personas in software development and criticized that personas are primarily considered only if the developers and designers are not working closely or if there is not customer available on site. Hence Getto (2014) argues that "just because individuals from different nations and regions can interact does not necessarily mean they will". Scenarios consists of information about what, when and why would customers use this specific API, whereas personas are built from users' needs, mental models and defining their expected experiences and skills. With scenarios and personas, it is a lot easier to define the actual requirements that customers have. (Dayley & Oliffe, 2017) The scenario describes the preferred outcome from where requirements can be extracted by analyzing the scenario, like in the example shown below in table 3.

Scenario	Requirements
External developer wants to create spare parts marketplace application for iPad	• Access the API outside of the company • Access the API from any web browser and device
When the app is published, it will allow customers to shop spare part and pay bills	• Pay bills • Read coverage and price information

Table 3. Extract requirements from scenarios

2.2.3 Requirements

When the business purpose, scenarios, and personas are clear, the next step is to decide API product features that will be implemented, and which shall be left out (Moilanen et al., 2018). The decisions are made based on planned benefits and value that is expected. Evaluation can be accomplished by asking simple questions: what are the business requirements? Do these requirements represent the business needs completely and clearly? For example, in the table 4 it is presented, if the objective is to allow students to view their results online, the business need would be improving access to information, and the business requirement is providing online access to information. Business value gained would be increasing the speed of process (Alnabhan et al., 2014).

It’s clear that prioritization for business requirements is needed to implement only vital features first and follow minimum viable product ideology. By separating functional and nonfunctional requirements a separation between quality attributes and needed functionality for customers can be accomplished. Functional requirements answer the question, “what user can do”. For example, “user can login” whereas non-functional requirements to answer the question, “how to do it”. (Alnabhan et al., 2014) For example, availability, scalability, logging or security (SOA Software, 2012).

Customers need	Business need	Business Requirement	Business value
allow customer to view their parts	improve access to information	provide online access to information	increase speed of process
allow customer to select and order parts	improve service decrease defects	provide online shopping cart	decrease number of employees decrease number of defects increase speed of process

Table 4. Part of List of requirements. (Alnabhan et al., 2014)

Prioritization is a tool, which aims to support decision making process. Distinguishing the valuable requirements from the trivial ones helps the key stakeholders to decide core requirements for system (Berander, 2007). Specifically, globally significant concerns that might have an impact on requirements that affect architectural setup or the whole interface design should be carefully evaluated (Duan et al, 2009). There are many techniques for prioritization, but it should always be discussed based on circumstances which one to use (Aasem, Ramzan & Jaffar, 2010). Some of the techniques are more sophisticated than others, which is complexity in the techniques. AHP and CV are rated from very complex to complex with fine granularity. In the contrary Top-ten, numerical assign and ranking are rated from easy to extremely easy with medium and coarse granularity. (Berander, 2007)

A general overview from the process of prioritization requirements starts from concentrating to users' needs, but also domain information, existing system information and different standards and regulations needs to be considered. After definitions an elicitation of requirements should be performed following analysis and negotiation with key stakeholders. In the next step requirements are documented and validated and furthermore after validation system specifications and requirements are agreed. (Berander, 2007) Otero, Dell and Otero (2010) evaluated requirements by ranking, based on quality attributes to determine the relative priority. They calculated an overall desirability for each requirement to provide optimal benefit vs cost value. Quality attributes for requirements were type (functional, imposed, product), scope (affecting many or all have higher priority), customer satisfaction (number of customers the requirement satisfies), perceived impact (PMF), APP specific attributes (usability, performance, safety, security, reliability and interoperability), and penalties (costly, risky, complex). These attributes are presented in depth in the chapter 3.3 and further analyzed in chapter 5.3.

The process for prioritization depends on the chosen technique and it's not relevant for this study to go through more than one example shown above. Users requirements change over time. Even if a software system fits perfectly to users' requirements, it still needs continuous optimization and measurements to keep up with the users' needs. (Fielding & Taylor, 2000) Budgetary restrictions and time to market constraints often dictate the need for stakeholders to prioritizing requirements carefully and collaboratively to recognize conflicting requirements and negotiate solutions before the actual triaging. Stakeholders need to evaluate security risks such as access control, availability, network load, integrity, data security, data location and data segregation. Stakeholders need to evaluate privacy risks such as international laws affecting service provider location, legal liability and incident management. And stakeholders also need to consider risks regarding the consumer with universal Terms of Service, program and privacy policies, and Copyright notices (Duan, Laurent, Cleland-Huang & Kwiatkowski, 2009).

2.3 API Design and Development

A huge amount of digital services come up to market every day. The first impression that customers get is the design; not only the interface but methods for solving problems. (Gebhart, Giessler & Abeck, 2016) API Design is a fundamental premise that API exists to serve a business purpose, even though it might be created for numerous reasons, the form and function should be driven by business requirements (SOA Software, 2012). Before design phase, there needs to be a planning phase which serves to organize goals, rules and procedures as well as requirements. (Papazoglou & Heuvel, 2006) It should enable easy access to try and use to avoid negative impacts like low adoption rates or looking into opponent's similar solutions. When a new product is being developed the whole development, team should focus on good design decisions instead of only focusing on functionality (Gebhart et al., 2016). Therefore, when speaking of API, it should be always made clear which context it is discussed in. Positioning helps organizations to target the right customers to invest into APIs. Developer-oriented positioning eases developers' work and presents capabilities that enables innovative opportunities for developers, whereas business partner positioning focuses on opportunities that APIs present for the business (Barnes et al., 2018).

Importing API from specifications, creating API endpoints, defining service contracts, and creating documentation, should be part of API design (Vijayakumar, 2018). All design decision should enable developer experience which can be ensured throughout the usage experience (CA Technologies, 2015). First questions that comes up to developers when getting into a new API are: What can be done with the API? How to get API up and running? What is the basic API architecture? And how to implement a use case (Meng, Steinhardt & Schubert, 2018) Therefore, these questions should be answered in the documentation as clearly as possible.

Identifying and specifying web services and business processes step by step is the goal of the design phase (Papazoglou & Heuvel, 2006). Only few organizations have experience from consumer-centric design processes, even though the value of usability is recognized. Creating consumer-centric APIs it is required to engage with the consumer throughout the process to gain feedback in early stage (Matheny, 2017). Delivering features that customers don't want can cause API providers to produce extra capacity causing delays and cost overruns (Ravichandran et al., 2016). CA Technologies (2015) define developer experience (DX) as interactions between API provider and developer, where the overall result is more a feeling about how did the interface make developer feel, whereas Vijayakumar (2018) extracts two types of developer experiences: technical experience and financial gain. By technical experience, Vijayakumar means documentation, onboarding process, and SDKs, and by financial gain commission strategy and advertising policies. This feeling is much harder to measure and detect than the functionalities used, which is why APIs should be designed in cooperation with the consumers (internal and external developers)

(CA Technologies, 2015). For example, even if APIs are designed in a simple and flexible way, developers might not engage with it if signing up is too difficult and time consuming. Pilot partners are early adopters who will provide vital feedback in the early stages of development, and if everything works as expected, the pilot can be used in promotion and marketing. (Bortenschlager, 2015)

Business beneficial B2B that APIs enable are increasing turnover by higher volume, customer loyalty and new innovations (Moilanen et al., 2018). Therefore, the movement is called API economy, which describes how to take advantage of exposing APIs as a part of services and expanding business model (Gamez-Diaz, Fernandez & Ruiz-Cortez, 2017). For example, if a company's business strategy is to gain more customer stickiness, or in the case of generating better market awareness, creating new sales channels, should be the priority when developing an API (SOA Software, 2012). Badly designed APIs with poor quality code will cause non-functional performance issues causing customer and revenue loss and furthermore negative impact to company’s brand (Ravichandran et al., 2016). The next step after API design is the implementation, but still there are some aspects to consider before implementing APIs, which are presented in figure 8. It is vital you should know the environment you are working with: it’s existing and planned APIs, services that could be leveraged, and high-level business requirements – does the API answer to the need? What is the urgency of delivery? And is it for external or internal use. (Soa Software, 2012) Underutilized resources and only partially completed work causes an increase in capital and operational costs (Ravichandran et al., 2016).

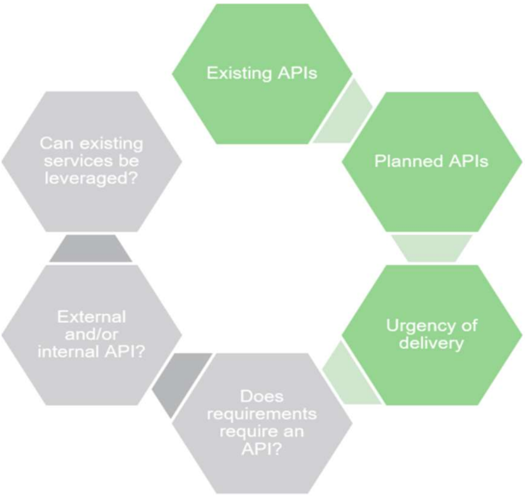

Existing APIs
Planned APIs
Urgency of delivery
Does requirements require an API?
External and/or internal API?
Can existing services be leveraged?

Figure 8. Before Implementation you need to know

2.3.1 Specifications

Defining structural, behavioral and policy specifications was the core methodology for service-oriented design and development (Papazoglou & van den Heuvel, 2006), and it still is in line with current studies in high levels even though more specific standards have evolved. For each API there should be API description, supported languages, SSL support, authentication model, data formats and other properties like sample source code (Fletcher, 2018). Limitations to functionalities, time, and operations, should also be described (Gamez-Diaz et al., 2017). Developers approach APIs with two goals: decide about whether the API should be selected in specific context for specific task or use the API to perform tasks. To gain answers to either of these cases, it should be made clear in the API specifications what can be performed with it, how to deploy it, and what is the basic architecture of the API since these are the most popular questions from developers when acquainted with new API (Meng, Steinhardt & Schubert, 2018). Specifications; questions that needs answering and decisions needed to make in API development are gathered into figure 9. to get a whole picture about the complexity and variety of aspects to investigate, which are further discussed below and finally concluded into table 5. as top 10 design principles constructed multiple sources, like Dayley's and Oliffe's guidance framework presented in appendix A and design EULA and TOS in appendix F.

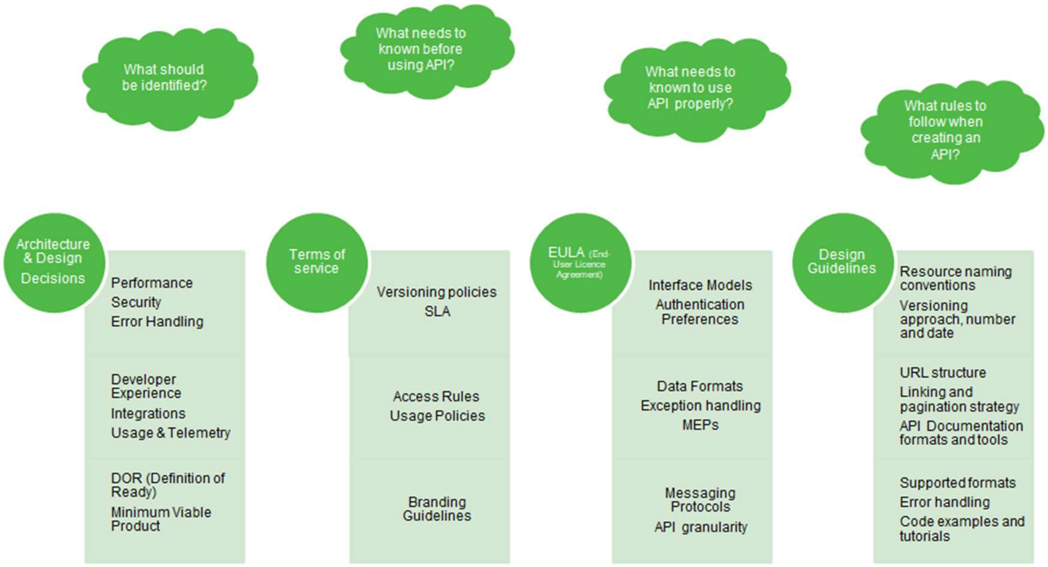

What should be identified?
What needs to known before using API?
What needs to known to use API properly?
What rules to follow when creating an API?
Architecture & Design Decisions
Performance
Security
Error Handling
Developer Experience
Integrations
Usage & Telemetry
DOR (Definition of Ready)
Minimum Viable Product
Terms of service
Versioning policies
SLA
Access Rules
Usage Policies
Branding Guidelines
EULA (End-User Licence Agreement)
Interface Models
Authentication Preferences
Data Formats
Exception handling
MEPs
Messaging Protocols
API granularity
Design Guidelines
Resource naming conventions
Versioning approach, number and date
URL structure
Linking and pagination strategy
API Documentation formats and tools
Supported formats
Error handling
Code examples and tutorials

Figure 9. API Specification Framework Derived from Vijayakumar, 2018; Dayley & Oliffe, 2017; Matheny, 2017; Murphy et al., 2017 and Boyd, 2015

Agile methods allow businesses to adapt quickly and detect problems early in the lifecycle which means less costs when correcting them. Velocity of the development and delivery can be increased by working parallel across development and testing (Ravichandran et al., 2016). DevOps aims to combine development and operations into a single seamless experience to respond even faster to customer's needs. When multiple people are working parallel to design an API, it needs to be very clear what has been already decided and done (Moilanen et al., 2018). Definition of done (DoD) is a set of criteria defining when a product is deliverable meaning minimum restrictions that needs to be fulfilled before release (Silva et al., 2017). Overall checklist should be the tool that helps developers audit and make sure which listed things are already implemented. It is a useful tool when buying, inheriting, or designing API related actions. (Moilanen et al., 2018)

Minimum Viable Product (MVP) was founded by Frank Robinson in 2001, being a product with the smallest amount of work possible satisfying customers' needs. Later on Eric Ries initiated the classification of MVP shown in figure 10. The goal of MVP is to make sure that customer need has been understood correctly (Moilanen et al., 2018) MVP can be a user interface like real world working product, but the business process is not automated or fully functional yet (Nguyen-Duc, Khalid, Bajwa and Lonnestad, 2019). Sometimes it might be wise to change direction if MVP shows that the customers problem hasn't been understood correctly (Moilanen et al., 2018).

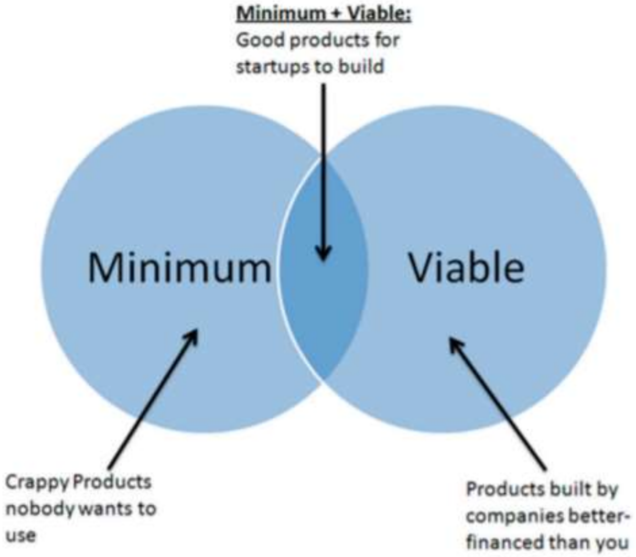

Minimum + Viable:
Good products for startups to build
Minimum
Viable
Crappy Products nobody wants to use
Products built by companies better-financed than you

Figure 10. Definition of Minimum viable product (Nguyen-Duc et al., 2019)

Terms of Service (TOS) clarifies to consumer what needs to be known before the API is used: access rules, versioning policies, branding guidelines, SLAs and usage policies (Dayley & Oliffe, 2017; Matheny, 2017). For customers, a TOS can also include a non-disclosure agreement ensuring that only wanted content assets are available via API (Boyd, 2015). Contracts about API consumptions should also contain considerations about usage; when, how much and what purpose is the API used for, branding; requirements for use and branding objectives, liability; reducing risks and ensuring availability and performance, data ownership; clarify who has right to the data flowing through and the data derived from the traffic as well as geographical concerns; and taking legal and practical constraints into account (Holley et al., 2014). End-User-Licence-Agreement (EULA) specifies how the API can be used properly (Dayley & Oliffe, 2017). This document should include information about which interface model, exception handling methods, used data formats, messaging protocols and exchange patterns are chosen along with granularity and authentication.

There are a few well known principles that should be considered when designing rules to follow, like Postel's law and backwards compatibility, but it still surprises me how different the rules are in practice when Murphy, Alliyu, Macvean, Kery and Myers (2017) presented an overview from 32 design guides about categories that was mentioned and further studied 10 of them: status code, response structure/format, standard methods, naming, versioning URI/URL structure, error response, backwards compatibility, documentation and custom methods presented in appendix C. Murphy et al. were able to come up with some commonalities that companies might take as a norm in general, which I shall discuss in following paragraphs. Similar guidelines were also introduced in Masse's Design Rulebook (2009) which consists of rules and practices for naming operators and parameters, versioning, error handling, security principals, used data models and methods (Moilanen et al., 2018). On top of these, Matheny (2017) lists that also examples, linking and pagination strategy, API documentation tools, and formats should be clarified. Documentation guidelines that affect customers decisions about API usage should be mentioned early on in the documentation as well as the information about what customer can or can't get (Robillard, 2009).

Naming in Murphy's et al. (2017) study was found to be separated to either follow "camelCase" or "snake_case" whereas response was consistently JSON, which is better for developing web-based applications (Gao, Zhang & Sun, 2011), or XML with varying time format from ISO-8601 to RFC-3339 standards. Most used URL/URI structures suggested that nesting shouldn't continue after one sub-resource (resource/identifier/sub-resource) as Matheny (2017) also rationalize it, enabling developers to speed up because of the machine- and human-readable features.

APIs should be able to extract events generated by separate user-operations (Lin, Chen, Xia & Sun, 2006), like GET, POST, PUT, and DELETE enables the usage. These are the standard methods along with suggested status codes: 200, 201, 400, 401, 404 and 500 with error response including error code, type and message (Murphy et al., 2017). OpenID Connect is most fit for APIs authentication needs (Naik & Jenkins, 2017), whereas OAuth and OpenID are both considered as standards (Dayley & Oliffe, 2017; Boyd, 2015).

Versioning should follow a few guidelines based on response logic; if it changes the way responses are handled it should be put in the URL for visibility, and if it doesn't it can be put in the header with maximum of three number semantic numbering (major.minor.patch) (Murphy et al., 2017). General guidelines for versioning strategy were made by Dayley and Oliffe (2017): previous and current versions should be supported at minimum before a version deprecates a fair warning of at least 6 months should be given and running two versions of same API should be avoided. Documentation should be generated automatically to machine-readable but also understandable for humans - Open API specification format, which was formerly known as Swagger (Matheny, 2017; Murphy et al., 2017). Open API specification (Swagger) is a Framework used to describe an interface based on REST architecture which is machine-readable. It describes, produces and visualizes the structure of the interface meaning that documentation and source code will be up to date at the same time (Haupt, Leymann & Vukojevic-Haupt, 2017).

Design Guidelines enables unified developer experience and lower learning curve for users. It will affect through the API lifecycle a decreasing workload and promoting maintainability (Chen, Xu & Zhu, 2012). Similar thoughts were brought up by Moilanen et al. (2018) pointing out the fact that Design Guidelines are a part of the developer communication network between customers, and internal development guiding technical and operational contracts that developers can rely on their own application processes (Oliffe, 2018). Below table 6. lists the best design practices from the literature.

Category	Suggested methods	Source
Status code	200, 201, 400, 401, 404 and 500	Murphy et al., 2017
Data structure / format	JSON; XML	Gao et al., 2011; Murphy et al., 2017, Dayley & Oliffe, 2017
Standard methods	GET, POST, PUT, and DELETE	Murphy et al., 2017
Naming	"camelCase" or "snake_case"	Murphy et al., 2017
URI/URL structure	resource/identifier/sub-resource	Matheny, 2017, Murphy et al., 2017
Error Response	error code, type and message	Murphy et al., 2017
Documentation	Open API specifications (Swagger); Mulesoft, API blueprint	Murphy et al., 2017, Matheny, 2017; Dayley & Oliffe, 2017
Security	OpenID; OAuth	Naik & Jenkins, 2017; Dayley & Oliffe, 2017
Time format	ISO8601 or RFC-3339	Murphy et al., 2017
Versioning	major.minor.patch in URL or header	Murphy et al., 2017

Table 5. Top 10 Design Guidelines for Developers

2.4 Build and run

At this point there should be a clear understanding of work goals, technical requirements and personal preferences of targeted developers. Before moving on to production API bound to real data or backend systems, a prototype should be built to test design assumptions based on target persona. With prototypes, it is much less costly to make changes and it takes less time to build it (CA Technologies, 2015). Each API should be managed independently, but within the same environment using centralized oversight and platform integration (Oliffe, 2017). The goal for API is to push work towards customers, for example if a customer requires customization for their B2B connections you have the leverage in the relationship to allow it (O'Neill & Golluscio, 2017).

APIs can be exposed in many ways, but the most developer friendly way is to create API platform because of its capabilities. It should include developer portal, sandbox, API framework and server platform, API security and management, option for on-premise deployment, or hybrid cloud/on-premise deployment and API PaaS (Oliffe, 2017). Developer portal acts as the entry point for developers to sign up, access and use API. Getting access to APIs should be made as easy and fast as possible (Bortenschlager, 2015). Integration between Developer portal and existing identity management infrastructure allows companies to support authentication and authorization via SSO (Single-Sign-On) for user identities already existing, provisioning processes and user registrations. To enable these capabilities organizations need identity federation using SAML (Security Assertion Markup Language) (Oliffe, 2017). API Platform gives organizations the ability to define and guide roles and processes to meet their standards (Soa Software, 2012).

Roles throughout API Lifecycle Management have different tasks. API Product Manager is responsible for managing the team, branding, marketing, costing, tracking and billing for the API (SOA Software, 2012), as well as identify user profiles (O'Neill, Malinervo & Dewnarain, 2017). Additions to these Holley et al. (2014) promotes that product manager should also define customer's needs, and directions for developers and define partner relationships. The developer's main responsibilities lie in the areas of development regarding to SDKs, API sandbox, security, logging and monitoring (non-functional capabilities). The technical writer is responsible for ensuring that the documentation is complete, accurate and accessible, whereas the community manager makes sure that the developers are deriving value from the community and the company itself by supporting the App developer community (SOA Software, 2012). API support staff is responsible for correct and smooth running of the API on top of managing and responding to trouble ticketing reporting directly to community manager (SOA Software, 2012).

Companies might think that if a customer API has been given customers to use, they automatically will, but that might not be the case. APIs don't create any value unless they are being used by applications. The role of marketing is to make it easier for customers to buy products, while strategic marketing companies provide the foundation that drive effectiveness of tactical marketing and go-to-market efforts (Barnes et al., 2018). Regardless Benzell et al. (2017) states that immediate increase in profitability may be seen when firm adopt API strategy, because it might be easier to sell or market existing products through complementary apps. In addition, of course good marketing makes it easier to build desirable products, design compelling experiences, and handle sales strategies in a repeatable manner (Barnes et al., 2018).

2.5 Monitor and Measure

Monitoring is essential in B2B API scene since it's supposed to form a central part of customer's application's functionality but there is established metrics to measure and understand API based on their interface (Bermbach & Wittern, 2016; Janet et al., 2014). Without detailed monitoring quality problems might not even occur, but instead show as decreasing usage rates (Bermbach & Wittern, 2016). Selecting suitable key performance indicators (KPIs) for API is highly important, since they are always contextual factors within value chain. Most common KPIs for APIs are number of developers, customers, partners, APIs and apps, as well as speed to API and onboard, traffic growth, business breadth, cost reduction direct revenue and NPS. (Leppitsch, 2018)

There are four dimensions and metrics that companies can use to measure effectiveness presented in figure 11 and further explored below. Each of the dimensions affect each other either enforcing them or decreasing their value. Customer and business value can be measured by NPS (net promoter scores), lead times, revenue per user story, or customer conversion (Ravichandran et al., 2016). Even though Benzell et al. (2017) point out that making assumptions based on the amount of API calls, it is not always that linear. It's entirely possible that high amount of API calls is the result of a poor performance, so APIs need to be called many times to get the wanted information. Similar notions have been made by Barnes et al. (2018), justifying that simple metrics, like traffic rates, are not always a good way of evaluating API usage, since it might just mean that developers have built a chatty client application. Instead of using simplistic metrics to measure benefits gained from API, business outcomes should be measured, for example, by grouping APIs together in reports to assess the overall value (Barnes et al., 2018). Thus, some business value can be measured directly from number of developers, apps and partners, traffic growth, business breadth, cost reduction and of course direct revenue (Leppitsch, 2018).

Efficiency and effectiveness can be measured by FTE to customer ratios, cost per transaction/app, or change/release cost burden (Ravichandran et al., 2016). Also, time-to-market from API idea to actual API is a useful measurement regardless of the complexity of the API. One simple way of measuring efficiency is the number of APIs, even though it's short-term and ignores easy-to-use utility and relevance of the API. (Leppitsch, 2018). Measuring quality and velocity might be a bit challenging, hence Ravichandran et al. (2016) introduces that usable tools are for example MTTR (Mean Time To Repair), rollback rates, cycle times, deployment frequency and operational support costs. Bermbach and Wittern (2016) studied specifically qualities of availability and performance distinguishing five measurable attributes:

1. pingability: responding low level request like ICMP protocol
2. accessibility: responding to HTTP request with predefined HTTP status codes
3. successability: responding with 2xx or 3xx HTTP status codes
4. latency: time between the start of the request and end of client receiving response
5. throughput: amount of API requests handled at given time

Culture, collaboration, and sharing are the least known and measurable metrics even through that is the basis for any other dimension in creating customer and business value. You should be able to configure a way to measure mentoring, open source contributions, job satisfaction and staff retention (Ravichandran et al., 2016). The usability of an API can affect collaboration, since if the documentation is poor, methods are named too similarly, or methods have too long parameter lists, which are hard to remember, it will decrease the usability (Rama & Kak, 2013). Also, the speed of onboarding process should be measured to see if all automated processes are working correctly (Leppitsch, 2018).

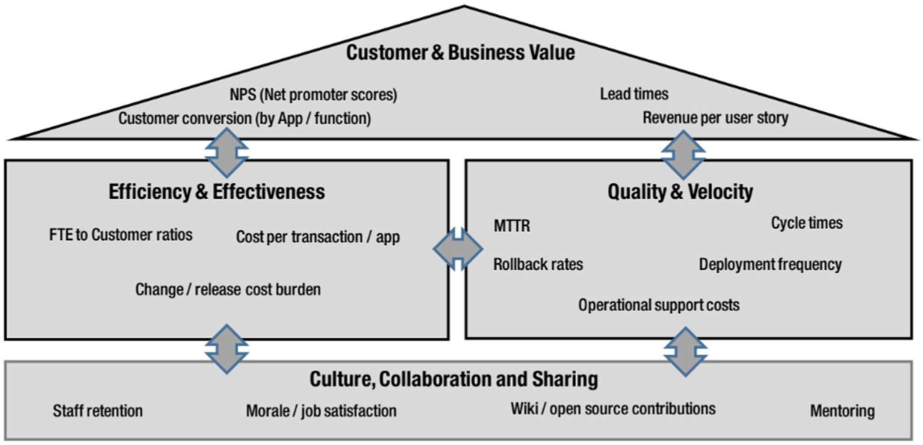

Customer & Business Value
NPS (Net promoter scores)
Customer conversion (by App / function)
Lead times
Revenue per user story
Efficiency & Effectiveness
FTE to Customer ratios
Cost per transaction / app
Change / release cost burden
Quality & Velocity
MTTR
Cycle times
Rollback rates
Deployment frequency
Operational support costs
Culture, Collaboration and Sharing
Staff retention
Morale / job satisfaction
Wiki / open source contributions
Mentoring

Figure 11. Effectiveness of DevOps initiative (Ravichandran et al., 2016)

Monitoring the APIs usage will give companies insight about money-making abilities, popularity, and if and what kind of extensions they should create for API. From performance, availability and consumers of an API company can measure most popular consumers, operations and overall consumption per consumer (SOA Software, 2012). Like, for example, Skandia, Tradeshift and Storebrand use metrics such as number of calls per hour and throttling limits to understand usage and control access (Boyd, 2015) and furthermore it is criticized that value might also be a lot harder to measure than simple response or call rates. (Barnes et al., 2018). When measuring the usage of APIs and the business benefits gained, it is vital to understand the overall picture about where and how APIs affect, to build correct meters for measuring the effectiveness of an API. Strategically measurements should be done in different point of views throughout the API lifecycle, such as business, developer, service provider, security and production points of view (Moilanen et al., 2018). Depending on the point of view, the metrics vary from simplistic usage measurements to more complex ways of verifying potential value created (Benzell et al., 2017).

Metrics for a developer portal are page visits, signups, API traffic, and support requests, whereas measuring the ease of deployment can be measured by TTFHW (Time to first hello world) (Bortenschlager, 2015). Measuring API can be performed also by including detailed analytics at the API layer which makes it easier to find the root cause, making overall analytics thorough and conclusive. Front-end latency, policy violations and routing failures, as well as real-time and historical analysis can be used to gain a clearer understanding on what API design improvements that are needed (Ravichandran et al., 2016).

2.6 Evaluate and Analyze business value

APIs deliver value relatively in complex ways, but a good developer experience is essential, whether talking about internal or external APIs, but a lot harder to quantify than functionalities (CA Technologies, 2015). Developers often have problems finding and accessing the information needed. A transparent information structure, efficient navigation and search options would help mitigate these struggles and ease the use for developers. Therefore, expertise of software developers, information design, and communication professionals are needed to satisfy the needs and expectations of the API consumers (Meng et al., 2018).

Stylos and Myers (2007) defined APIs basic qualities to be its usability and power. By usability they refer to qualities like how easy API to learn, how productive, simple and consistent it is, and how well it matches users' mental models. By power, they refer to qualities like APIs expressiveness, extensibility, evolvability and performance. Fletcher (2018) determines three elements of API quality: functionality, reliability and usability presented in figure 12. The most important quality attribute for web API is its usability. Usability is defined "as the degree to which a product or a system can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in a specified context of use". (Bokhary & Tian, 2018) Usability of a web API can be evaluated with a black box approach, which means the understandability of documentation; how it can be learned and functioned with. Reliability is based on the maturity level of API and measured by considering available statistics of usage components along with frequency of its changes and updates (Fletcher, 2018). API functionality consist of three elements: interoperability, compliance and security. Interoperability depends from the APIs capability to be used in different and heterogeneous environments, whereas compliance becomes better when API support more standard data formats, which increases overall interoperability. Security of an API needs to consider at least SSL support and authentication (Fletcher, 2018).

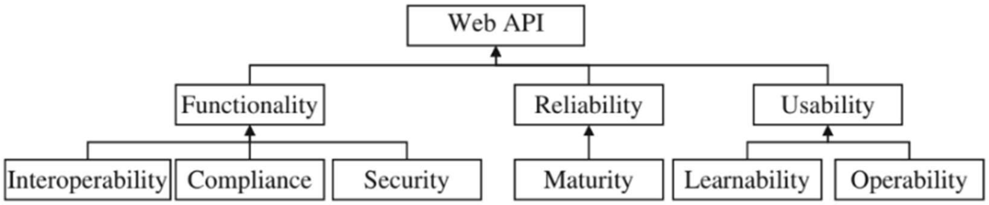


Figure 12. Quality model for mash up components (Fletcher, 2018)

Fletcher's API quality model is based on Cappiello, Daniel & Matera (2009) from Mash up quality model. It seems that this model is lacking some perspective regarding what the API can do to increase company's overall viability. Therefore, I expanded Fletcher's model with parts from ISO/IEC 9126-1 standard for Software quality presented in figure 13. To decompose quality model further into measurable elements the IOS/IEC 25010:2011 could be used along with Bokhary's and Tian's (2018) method for measuring cloud service API quality and usability presented in appendix E. It is not relevant for this research and therefore it's excluded.

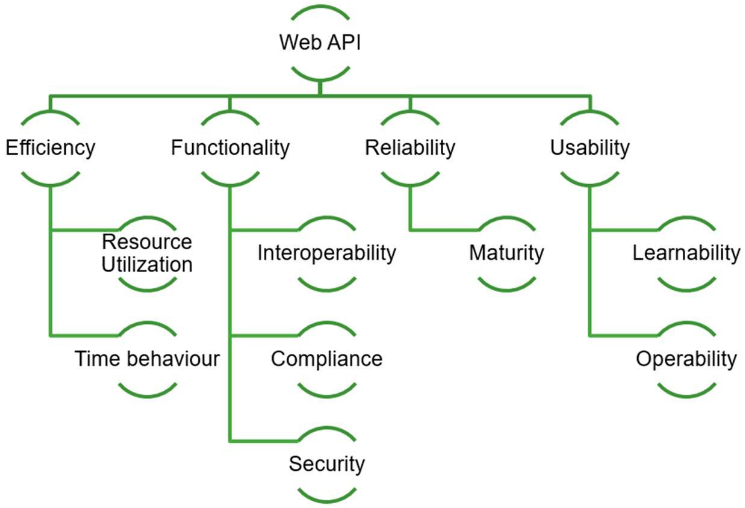


Figure 13. Quality model for Web API

It should also be ensured that chosen API management solution supports the monetization of APIs to build enough flexibility into API pricing model. Revenue can be generated indirectly and directly from the same API if the flexibility allows it (O'Neill, Malinverno & Dewnarain, 2017). The most common monetization model is indirect, where new capabilities are increasing "stickiness", but if the API returns highly valuable data for the consumer, direct monetization methods should be considered: pay per use for developers, subscription for enterprises or revenue share for partners (Barnes et al., 2018). Negative brand impact could be caused by badly designed and poor-quality code, which creates non-functional performance issues leading customers to abandon the company and eventually losing revenue (Ravichandran et al., 2016).

APIs direct impact to company performance can best be seen in net sales only internally, whereas external impact is seen in market share and valuation. In many cases the amount of API calls or users do not correlate straight to net turnover, but more as customer loyalty and satisfaction (Benzell et al., 2017). Simple statistics like traffic rates and response times from API calls might not be a good measurement for API usage, since it might only inflict that a "chatty" client application has been built by developers (Barnes et al., 2018). Overall the methods used to evaluate APIs depends highly on the entire digital value chain (Leppitsch, 2018).

2.7 Support

A good developer experience can be defined based on the interactions between API provider and the developer, but in the end it all comes down to how does the interface make the developer feel (CA Technologies, 2015). Businesses risk substantial losses from their software investments if customers emotional and physical needs with their behavioral patterns are not understood (Ravichandran, et al., 2016). Poorly designed API can have consequences which impact is long-reached including higher maintaining costs, loss of revenue, frustrated users and missed business opportunities. The less you demand adoption from developers the less they need support and struggle to understand the use of the API (Dayley & Oliffe, 2017).

Most obstacles in documentation were unclear structure and navigation, complex document structure, and incomprehensible language (Meng et al., 2018). Therefore efficient, reliable and easy to deploy documentation is the primary key to support customer (Moilanen et al., 2018), since the top 5 important properties of good documentation were that it's up to date, searchable, complete, includes code examples, and explanations for classes and methods. The first steps that developers take when acquainted with new APIs are working through code examples, reading "getting started" document and overview with sample scenarios. It should be noted that most typical problems when getting into a new API were wrong, incomplete or incomprehensible documentation, integration problems with API, and lack of examples (Meng et al., 2018). Some of the best supporting practices have been listed below by Soa Software (2012), O'Neil et al. (2017), and Dayley and Oliffe (2017):

1. Recognize and interact with API developers
2. Continued customer support through clear communication channels
3. Convenient tools with self-service developer portal
4. Create SDKs
5. Documentation and how to use -guides
6. Enable easy testing
7. Flexible monetization (Developers, Enterprises, Partners)

2.8 Summary

The literature review provides theoretical foundation connecting API design principles to different stages of API's lifecycle. First by providing a context to B2B field and thereafter, investigating API design process from defining business need to be creating scenarios, defining API specifications, introducing roles needed supporting APIs life cycle management and how to measure and evaluate APIs usage and value. Some limitations and interesting further investigation topics of the previous literature are also presented along the literature review. Overall the literature review adds multiple theoretical components into a structured guide from higher management level to technical details.

3 Research Framework

The theoretical research framework is based on the literature review. Thus literature did not cover all the situations possible in a big manufacturing companies, which lead in the development of the API exploration model. This model sets out to understand how companies, that don't have business objective for API, nor clear business need defined.

To gain an understanding about which APIs to implement, it's required to have a need. The source of information in this study is the key stakeholder that is directly or indirectly in contact with the customer and therefore might have insight into customer's needs. Each key stakeholder is inclined with products and services, that might be potential API candidates. Each API candidate should be explored; if value is offered to customers. Finally, API ideas will be listed, analyzed and prioritized in API exploration process presented in figure 14. This process has been divided into three individual research models to keep it more understandable: discovery, analysis and prioritization.

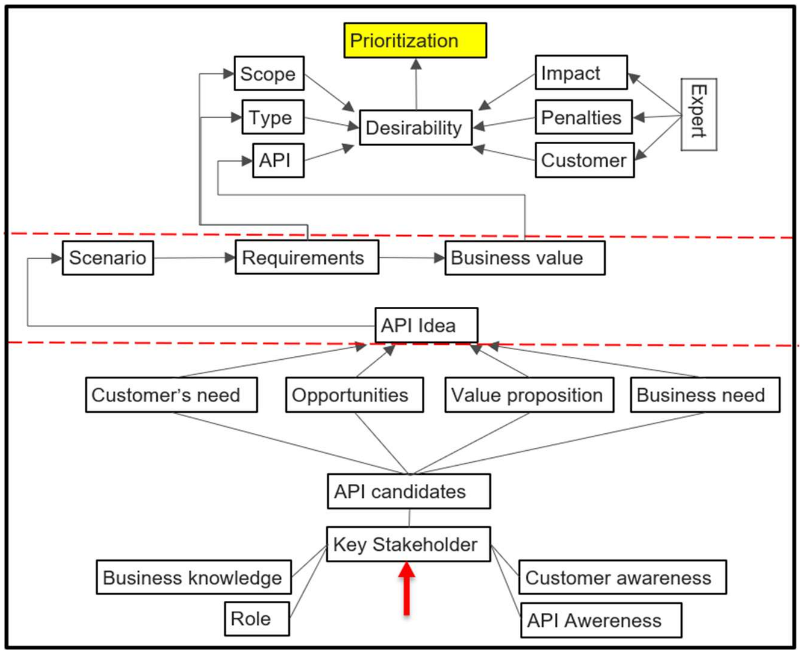


Figure 14. Research model for API exploration

3.1 Research model for API Idea Exploration

Often it is not possible to involve customers directly so an alternative way for discovering the business needs is defined. Research model for API idea exploration aims to discover key features in a specific business area to form beneficial API Ideas. The attributes to investigate for key stakeholders in the first phase are business knowledge, role, API awareness and customer awareness. Key stakeholders are involved with multiple services and products in the organization or company. Model is based on Moilanen et al. (2018) method to develop new APIs, which starts by identifying the API need with key stakeholders like salesman, customers or other relevant stakeholders. The goal of the second phase is to evaluate the API candidate by exploring why this product or service is needed by business or customer, what kind of benefits it's offering and what type of possible flaws or problems it might have. Based on these attributes the API Idea should be revealed, like figure 15 presents.

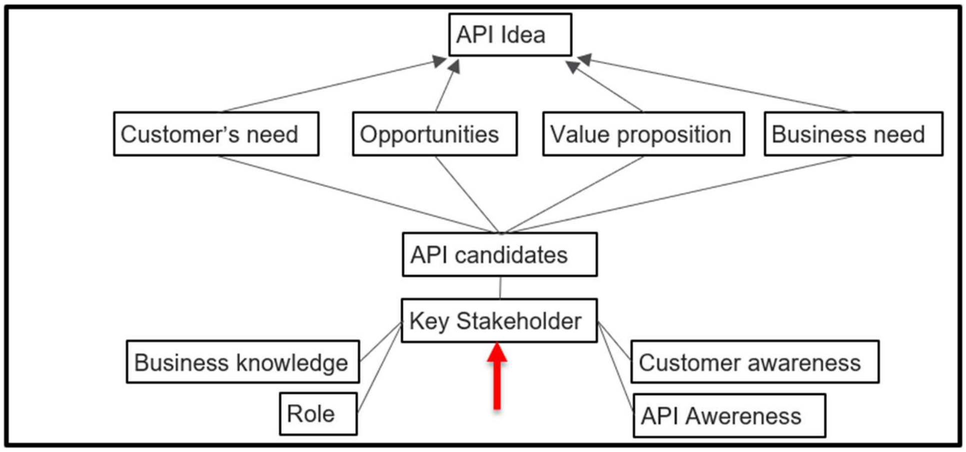


Figure 15. Research model for API Idea Exploration

KEY STAKEHOLDER

Key stakeholders are employees from which the organization's business draws resources. API should only exist to serve a business purpose and be driven by the business requirements derived from customers (SOA Software, 2012). Because it's often not possible to include customers, alternative resource of information for defining API Ideas can be used. Not all API candidates are able to use APIs efficiently, but to find the ones that can, it's crucial to first select the correct key stakeholders that can acts as informants to get the interviewer to get overall context understood as co-constructive sensemaking process (Kangas et al., 2019). Furthermore, selecting the correct key stakeholders is also essential to define customer needs and business needs for API ideas. Selection can be executed by considering following attributes: business knowledge, role, customer awareness and API awareness. With these attributes interviewer gains understanding of the knowledge level in the scope of organization and API to select valid key stakeholders. Finally, the overall understanding can be reshaped by exploring variety of explanations and reframing relations to real life context. (Kangas et al., 2019)

ROLE

Role defines the influence and power the key stakeholder has in the company they are working for as well as the tasks and responsibilities. Moilanen et al. (2018) started collecting information about a new API idea by talking with salespersons, account managers or customers about their needs for API. Because this might not be always possible in a large company thus I extend the model of Moilanen et al. (2018) and included alternative way, where the role of the key stakeholder could be evaluated as a parameter and all the key stakeholders could be included if needed in the study regardless of their role in the organization. The role of key stakeholders is essential basic information to gain to understand the position and angle the key stakeholder is looking the whole organization. The role of the key stakeholder impacts how wide visibility the key stakeholder has in the case organization and furthermore might impact the API ideas derived. When selecting the key stakeholders wanted level of visibility should be considered. The higher level in the organization a key stakeholder has, the wider the view is and vice versa the lower the level in the organization, the deeper view the key stakeholder has. The assumption is, that depending of the visibility, the key stakeholder's ideas might have different scale, where APIs should be used and further what kind of API ideas will key stakeholder form up.

BUSINESS KNOWLEDGE

Business knowledge aims to explain the amount of information and the period the key stakeholder has across the organization. Even if the key stakeholder has the aimed role for the APIs to be discovered if the key stakeholder only has a little experience from other sectors in the company or has been working only for a small period, the wanted results might not be accomplished. The more the key stakeholder has knowledge of the company, the better information can be explored during the interviews. Because APIs are often built to leverage existing product or a service, the knowledge seems utterly important to be able to any beneficial API ideas (Wulf & Blohm, 2017). Even though the key stakeholder does not have a clear idea what API is or how it could be utilized, this kind of key stakeholders might be vital to interview to gain knowledge from the organization, processes, services and limitations overall. Because in today's digital world it's important to think and develop new ways to operate in service centric organizations where communication is the key (McPhee et al., 2017).

CUSTOMER AWARENESS

Customer awareness inflicts how well the key stakeholder could put oneself into customer's position and wonder what case company's customer could need and want (Moilanen et al., 2018). Developing personas from demographic, contextual, behavioral and attitudinal data it would be necessary to perform surveys, ethnographies and interviews to find out what the customers want or need (Getto, 2014). When these methods cannot be used, customer awareness can be used to measure how qualified the information being received from the interviewee is. Customer awareness depends a lot from the fact that if the key stakeholder works closely with the customer or not but also from the individual's skill to put oneself into the customer's shoes. To create valuable APIs, it's potential users' aka customers should be known, since most of the profitable API Ideas comes directly from the customer (Moilanen et al., 2018). Therefore, customer awareness is a key element, when the customer itself is not providing the information about which APIs they might need. This means discovering information about how inclined the key stakeholder is with customer; are they acquainted daily. Customer awareness inflicts how well the key stakeholder could put oneself into customer’s position and wonder what case company’s customer could need and want. Customer awareness depends a lot from the fact that if the key stakeholder works closely with the customer, but also from the individual’s skill to put oneself into the customer's shoes.

API AWARENESS

API awareness exposes quality and amount of knowledge that the key stakeholder has from APIs; whether the key stakeholder knows what it is and how well? This information or lack of it, might affect the quality of API ideas. Moilanen et al. (2018) went through possible connectors, plugins and interfaces with the selected key stakeholders to explore API needs. if this sort of information is not received, the evaluation further in the research might be lacking. Researcher should be able to reframe key stakeholders’ experiences and form their own understanding of the context, APIs. This understanding can be reshaped by exploring alternative explanations. (Kangas et al., 2019) Therefore the API awareness should be evaluated to understand the quality of the ideas pursued from the key stakeholder.

API CANDIDATES

API candidates are services or products that might benefit from APIs potentially. At high level APIs are exposed for three different reasons: exposing business assets or services to third party to unlock business value, exposing software interface that exposes business assets over the web to other systems or people and exposing HTTP endpoints for requests. (Janes et al., 2014) Products and services that usually yearn for API are those which needs online access to catalogs and inventory, expose core functions to external users, provide self-service options, build community, drive traffic or refactor business processes (Soa Software, 2012). Especially catalogs be good API candidates (Boyd, 2015) In order to find if the key stakeholders are working with such a product or service, knowledge should be gathered from the products and services that key stakeholder is responsible for aka API candidates.

OPPORTUNITIES

API opportunities are born when a certain API candidate has challenges in one or more areas like efficiency, functionality, usability or reliability. Knowledge about challenges should be gathered to form API opportunities. (Chapter 2.9) API might be the opportunity to solve a challenge for a certain product or service (Moilanen et al., 2018) For example efficiency as an opportunity refers to time behaviour on deeper level, meaning for example time used for certain task being replaced by automation. In this case there is an opportunity born due to the occurring challenge to make a service or product more efficient. Identifying these API opportunities requires often case-by-case communication at first, whereas later similar use cases should be looked for along with common datasets to scale for additional partners. (Boyd, 2015)

VALUE PROPOSITION

Every API candidate has a specific area they bring value into, which is value proposition in this research. To define whether API can bring more value to the service or product it should be analyzed what are the benefits for the customers. Those value adding components should be determined and dismembered like Moilanen et al. (2018) suggests. Next step is to determine value chains for API enabled solutions, since value is not coming directly from APIs, but from the applications that eventually use APIs. (Barnes et al., 2018) Value that APIs inflict could be for example lowering the complexity of the experience by standardizing, enabling easier access to data or offering a new channel for data offerors'. (Smith, Ofe & Sandberg, 2016) In business level API benefits be business benefits like volume, cost savings, customer loyalty and new innovations or startups (Moilanen et al., 2018).

CUSTOMER'S NEED

Customer's need can be defined as the reason why the customer is interacting with a company in the first place, there is a need. Defining Customers' needs without encountering with customer as a starting point is challenging. Without firsthand knowledge, you must rely on second hand knowledge from the key stakeholders about customer's needs. Another way of defining customer's needs is to gain knowledge about key stakeholders' thoughts if they were the customer, aka customer awareness presented earlier in chapter. (Moilanen et al., 2018) APIs that enable communication and sharing resources are extensions for business. In the global business environment access to real time data, like product information or order tracking are often low-risk datasets to open, that customers generally find helpful to manage their own supply chain better. (Boyd, 2015) If the API Idea makes existing tasks easier, more efficient or otherwise usable, it's ensured that the customer will want to explore their opportunities in API field as well, if not there already. Even though it might be beneficial to weight up, to which customer data should be offered. After all, it's always a strategic decision with a contract, which should not be made hesitantly. (Moilanen et al., 2018)

BUSINESS NEED

Business need is the reason why a new system, platform, or API is needed (Alnabhan et al., 2014). The business need for APIs can be anything from growing customer base by attracting customers to improving time-to-value and time-to-market for new services and products. APIs drives innovative ideas by capitalizing internal, external and open APIs. (Holley et al., 2014) More specifically Dayley & Oliffe (2017) determined a list of core functions, why APIs should be implemented: enable web and mobile applications, integrate internal applications, offer interface with microservices, publish data, create cloud and SaaS integrations, enable IoT interactions, engage customers and extend business. Whenever there is a customer need, there also needs to be a business need before it's beneficial to start the action. Business needs that should be considered are categorized as efficiency, functionality, reliability and usability described in more detail in chapter 2.9. and further decomposed as Bokhary & Tian (2018) presented, if possible.

API IDEAS

API ideas are description of the need for API or other explanation that makes sense of the specific use case; the more specific the better. Ideas about what could bring value to the customer and to the case organization can be formed based on previous knowledge, unless the idea pops out directly from the information gained. Ideas should describe a brief scenario or requirement(s) what the API would do, what data or functionality it would handle and the source and destination(s) of the data. The more information gained during previous steps, the better and detailed idea can be formed and the more accurate the analysis results will be. Every API idea should have a similar goal to push work towards clients (O'Neil & Golluscio, 2017). API ideas can vary based on their maturity level from digital service that digitizes, ensures safe use and availability of transactions to a whole API ecosystem. In between there are partner integrations that open new capabilities and partner infrastructure meaning an API portal or marketplace to create new use cases. (Ravichandran et al., 2016) API Idea can be formed from a challenge or a problem in a system. For example, a system, where customer needs to login from multiple windows to use all the features and software, an API repairs by providing easier access via single-sign-on. API Idea can also be a new feature -API Value proposition in a platform, for example customer portal. To offer better communication a chatbot redirecting support issues faster to correct employees might increase customer's usability and efficiency in customer portal.

3.2 Research model for API Idea Analysis

Further analysis for the API idea can be performed as the figure 16. shows. The process starts with forming a detailed scenario where the API would be used in a specific real-life context. When the context is formed, requirements can be extracted one by one. Requirement describes a single task that the API needs to enable. After requirements have been found, each requirement should be further analyzed to find the business value it's inflicting. More information presented in the chapter 2.6.

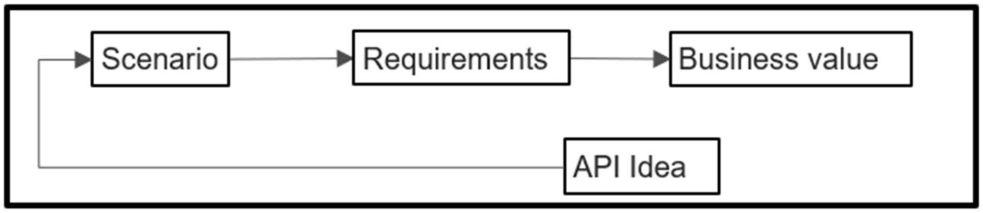


Figure 16. Research model for API Analysis

API IDEAS

When all the API ideas are gathered, they should be gone through one by one, analyzing the results from the API exploration phase. API ideas consists at least of the description about wanted functionality and a business need or customer need for wanted functionality. It might also include knowledge about the source or destination of the data or functionality as well as other requirements. Because the nature of this study is exploratory, a certain freedom should be given, when obtaining knowledge to gain a wide range of knowledge. Even just a described problem in API candidate could be used as an API idea at this point. Whether it turns into an API idea, will be revealed during the analysis phase. First of API idea should be analyzed, whether it fits into the given business needs. In this study I will use Dayley and Oliffe's (2017) API functionality list presented in chapter 2.4 to determine the business need: enable web and mobile applications, integrate internal applications, offer interface with microservices, publish data, create cloud and SaaS integrations, enable IoT interactions, engage customers and extend business. If the API fulfills one or more of these functionalities, you can proceed to the analysis to determine the scenario.

SCENARIO

Scenarios consists of information about what, when and why would customers use this specific API, with scenarios it is a lot easier to define the actual requirements that customers have. (Dayley & Oliffe, 2017) At first the description of the API might act as a scenario describing at a high level what the API does and who will use it. To determine the scenario further, either more specific knowledge should be gathered, or deep analysis should be performed towards the API Idea extracting the needed information. Scenarios can be used during the analysis to help define non-functional and functional requirements for the API. If the scenario cannot be identified, the API idea itself can act as a scenario. This might also inflict in fact, that the given API idea was already a scenario of APIs use case to begin with.

REQUIREMENTS

Requirements describe the rules for APIs implementation. By separating functional and nonfunctional requirements, a separation between quality attributes and needed functionality for customers can be accomplished. Functional requirements answer the question, "what user can do". For example, "user can login" whereas non-functional requirements to answer the question, "how to do it". (Alnabhan et al., 2014) Requirements can be extracted from the business need or from the API idea, which are formed as a scenario describing what are the actual actions that should be able to perform with the API. Extracting requirements from API Ideas starts by identifying the requirements that a user must use the API. For example, if the API Idea is "ability for external developers to create spare parts marketplace", the requirements would be at least that the API must be accessible outside the organization and with every browser and device. (Dayley & Oliffe, 2017) In this study ISO/IEC 9126-1 standard is used for defining non-functional requirements and functional requirements. These will be gathered based on the information gained form API exploration.

BUSINESS VALUE

Business value indicates what kind of value is the customer gaining by using specific API. If the objective is to allow students to view their results online, the business need would be improving access to information, and the business requirement is providing online access to information. Business value in this scenario would be increasing the speed of process (Alnabhan et al., 2014). By going through the requirements from the previous step one by one it's possible to extract customer's need, business need, business requirement and finally business value like presented in the table 6.

Customers need	Business need	Business Requirement	Business value
allow customer to view their parts	improve access to information	provide online access to information	increase speed of process
allow customer to select and order parts	improve service decrease defects	provide online shopping cart	decrease number of employee decrease number

			of defects increase speed of process

Table 6. Defining business value (Alnabhan et al., 2014)

3.3 Research model for API Prioritization

In this study I expanded and modified the prioritization model from Otero et al. (2010) to fit into API field as well, since it was stated in the study that this prioritization model is valid and can be further explored and expanded to other fields than software project requirements. The meaning of this model is to evaluate requirements by ranking, based on quality attributes to determine the relative priority by calculating the overall desirability for each API to provide optimal benefit vs cost value. The purpose of the model is to provide a requirement scheme for prioritization. The scheme represents how well requirements meet quality attributes. (Otero et al., 2010) Every API idea's attributes listed below will be analyzed with binary input evaluation method. After all the API ideas have been analyzed and the overall desirability has been calculated, the final prioritization can be performed based on the desirability percent. The process is presented in the figure 17.

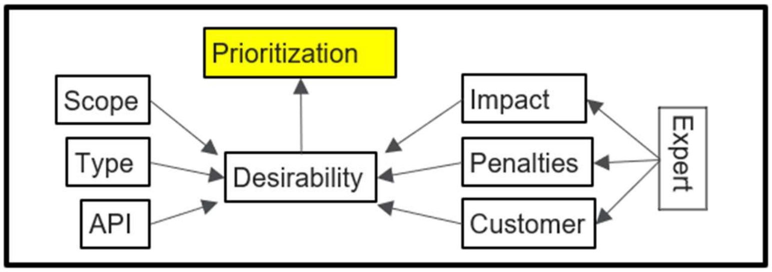


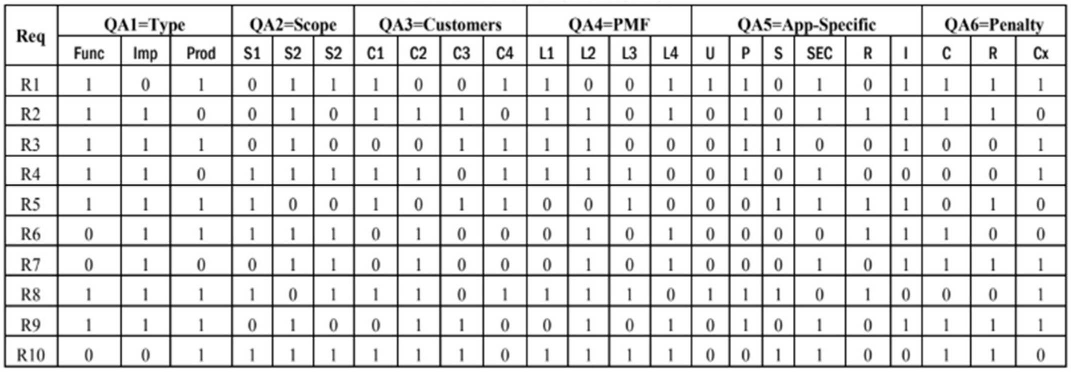
Figure 17. Research model for API Prioritization

Quality attributes used in this study uses weight 1,0% unless otherwise mentioned in the table 8. below. Each parameter's lower boundary was set to 0 and the higher boundary to 100. Distinguishing from Otero et al. (2010) customer parameter weight was lowered to 1, since in the case study does not contain first-hand knowledge from customers. If the higher weight (5) would have been given, misleading or flawed results might be formed. Target for each parameter was set to 100, except API and penalty. API was set at 70, since it's unrealistic to suspect that all the quality attributes would be effective at once. Penalty was set to 0, since the target is not to have the APIs classified as costly, complex or risky.

Parameters	Benefits					Cost
	Type	Scope	Customer	PMF	API	Penalty
Lower	0	0	0	0	0	0
Upper	100	100	100	100	100	100
Target	100	100	100	100	70	0
Weight	1	1	1	0.9	0.7	1

Table 7. Target values and parameter weights

Attributes to consider are presented in the table 8 below and weights for each are in the table 7. Binary input evaluation starts by defining the type (functional, improvement, product), scope (one, few or many subsystems), customer satisfaction (one, few or many customers satisfied)) and penalties (costly, risky, complex) with 1.0% weight. Whereas perceived impact by case company expert (PMF) with 0,9% weight and APP specific attributes with 0,7% weight: usability (operability and learnability), reliability (maturity), functionality (security, interoperability, compliance) and efficiency (time behaviour and resource utilization).



Req	QA1=Type			QA2=Scope			QA3=Customers				QA4=PMF				QA5=App-Specific						QA6=Penalty		
	Func	Imp	Prod	S1	S2	S2	C1	C2	C3	C4	L1	L2	L3	L4	U	P	S	SEC	R	I	C	R	Cx
R1	1	0	1	0	1	1	1	0	0	1	1	0	0	1	1	1	0	1	0	1	1	1	1
R2	1	1	0	0	1	0	1	1	1	0	1	1	0	1	0	1	0	1	1	1	1	1	0
R3	1	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	1	0	0	1	0	0	1
R4	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	1	0	1	0	0	0	0	1
R5	1	1	1	1	0	0	1	0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	0
R6	0	1	1	1	1	1	0	1	0	0	0	1	0	1	0	0	0	0	1	1	1	0	0
R7	0	1	0	0	1	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	1	1	1
R8	1	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1	0	1	0	0	0	1
R9	1	1	1	0	1	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	1	1	1
R10	0	0	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	0	0	1	1	0

Table 8. Binary Input evaluation (Otero et al., 2010)

SCOPE

Scope refers to the amount of user pools will be able to use this specific API. There are three different levels of binary points to gain: internal -gives one binary point, external - gives a second binary point and if the API is used by internal and external users, it will gain three binary points, which is the maximum aka 100%. The target for this attribute is 1, meaning favoring results with internal and external users in scope. If the API idea is internal it will gain only 33% of desirability, since in this study we aimed to search for B2B APIs as a priority. If the API is for externals only, it will gain 66% of desirability and will be prioritized higher when all the attributes are calculated together.

TYPE

In this study the type represents in a way the amount of effect the API has. There are three levels of binary points to gain: functionality -gives one binary point, improvement - gives second binary point and a new product will give the third point, which is the maximum desirability 100%, when the target is being 1. Functionality means a simple action that can be added into a system or application, which is needed whereas an improvement is either already existing feature, that can be effectively be improved by API or it’s usability can tremendously increase by API. API being a new product means that it’s completely new business need to fulfill and there is not such functionality existing now.

EXPERT -PMF

PMF (perceived as major functionality) considers whether the API idea seems relevant in the case company's point of view. In this study there are three different angles to look at: system, hardware and software. Whenever an API seems to offer relevant functionality that touches system, hardware or software -a binary point is gained. I decreased the weight slightly to 0.9, simply because I thought it would not be crucial to hit all of them. Because I wanted to stay true to the original model I did not tune it more, but it could be dropped down to 0,7.

CUSTOMER SATISFACTION

Customer satisfaction refers to the number of customers the API influences. In some cases, this information might not be available thus a substitute might be used to put themselves into customers shoes. This individual should have at least a touch of customer knowledge in the case company. This attribute has three levels, each gaining one binary point on top of each other in the following order: one, few and many. One gains 33% percent desirability, few 66% percent desirability and many 99% desirability. This means that if an API is very beneficial to only a small group of customers, it would gain 66% percent desirability. Whereas if an API is just a point to point integration towards the biggest client, it would only gain 33% percent desirability, even though this might be necessary and very much needed integration. Thus, if the API is small functionality available to all customers, it will gain 99% percent desirability. If the actual API idea was internal the analysis should be based on whether all the customers will benefit from the internal change or does it only affect smaller proportions of customers.

API

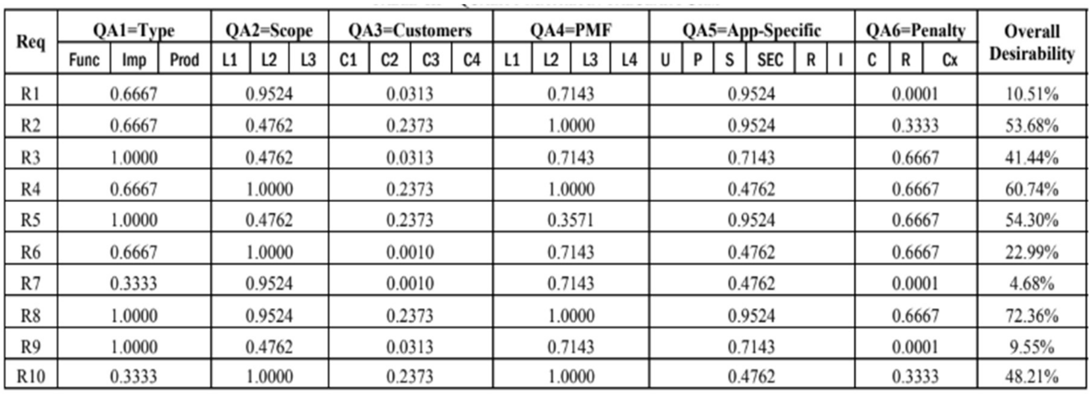
API attributes are being concluded from the business value gained. Each API attribute gains one binary point and the target being 0.7, not all of them are expected to be fulfilled. Usability is divided into two different categories: operability and learnability. Operability - binary point could be gained by offering new ways of operating and overall increasing usability in the point of actual act whereas learnability point could be gained when the API improves the way something can be used and learned. Reliability refers to maturity, which means the overall state of services, how sophisticated are they: are they just simple integrations that transfer data or are they an ecosystem that drives customer to use case company's services and platforms daily. Functionality is divided into three categories: security, interoperability and compliance. Security includes all the functions that improves security and access management, whereas interoperability refers to the ability to work with as many services and products as possible without problems. Last functionality category, compliance means actions that corrects or makes sure a thing is as it's supposed to be. Final attribute is efficiency, which is divided into time behaviour and resource utilization. If an API changes the way existing pipelines or workloads move faster, it means that the time behaviour is changed to more efficient one. Resource utilization means the ways the resources are used; if they can be used for example in 50 services instead of just one, it can be said that resource utilization is more efficient. Each of these attributes categories can gain one binary point. The target is 0.7 since it's not realistic to hit all of them in one API.

PENALTIES

In large industry setting penalties are costly, risky and complex. If an API will take a huge amount of money to make, it be a defect, since it's always a gamble at first, whether to gain users or not. If a lot of money is wasted, it's not the most efficient way to start API economy in a company. The meaning behind this is to find the most beneficial, easy and cheap APIs to implement first to gain users and later expand the field. Sometimes API can be too risky to create, it might have some technical issues or data being out of date. Too complex API Ideas refers to a situation, where easier approaches might be more efficient to create and manage. API is not always the correct way to go, so it needs to be evaluated.

DESIRABILITY

Before desirability can be calculated, the requirements need to be defined. These requirements have been defined in the analysis phase. Evaluation of the requirements quality provides the optimal benefit vs. cost value for an API. Desirability function can be used as a unified measurement that characterizes the quality of requirements based on predefined API requirements. Desirability should be calculated based on the binary points each attribute gained described above. After the binary points have been calculated, the weight should be applied and finally each attribute's desirability should be calculated together. The higher the desirability percent is, the more likely it is that the specific API will be valuable for the company inflicting value. Example of the desirability calculations in table 9 below.



Req	QA1=Type			QA2=Scope			QA3=Customers				QA4=PMF				QA5=App-Specific						QA6=Penalty			Overall Desirability
	Func	Imp	Prod	L1	L2	L3	C1	C2	C3	C4	L1	L2	L3	L4	U	P	S	SEC	R	I	C	R	Cx	
R1	0.6667			0.9524			0.0313				0.7143				0.9524						0.0001			10.51%
R2	0.6667			0.4762			0.2373				1.0000				0.9524						0.3333			53.68%
R3	1.0000			0.4762			0.0313				0.7143				0.7143						0.6667			41.44%
R4	0.6667			1.0000			0.2373				1.0000				0.4762						0.6667			60.74%
R5	1.0000			0.4762			0.2373				0.3571				0.9524						0.6667			54.30%
R6	0.6667			1.0000			0.0010				0.7143				0.4762						0.6667			22.99%
R7	0.3333			0.9524			0.0010				0.7143				0.4762						0.0001			4.68%
R8	1.0000			0.9524			0.2373				1.0000				0.9524						0.6667			72.36%
R9	1.0000			0.4762			0.0313				0.7143				0.7143						0.0001			9.55%
R10	0.3333			1.0000			0.2373				1.0000				0.4762						0.3333			48.21%

Table 9. Overall Desirability (Otero et al., 2010)

PRIORITIZATION

Prioritization can be made directly from the desirability percent, like Otero et al. (2010) did. Depending on the case company's needs, some weight could be added for example towards costs or complexity after the desirability have been calculated. In the figure 18. below there are costs and desirability chosen as an example. This prioritization framework will help the key stakeholders and managers above them to make decisions about which APIs should be implemented in which order. By evaluating the overall desirability and being able to predict for example the cost level it should be a straightforward decision to make.

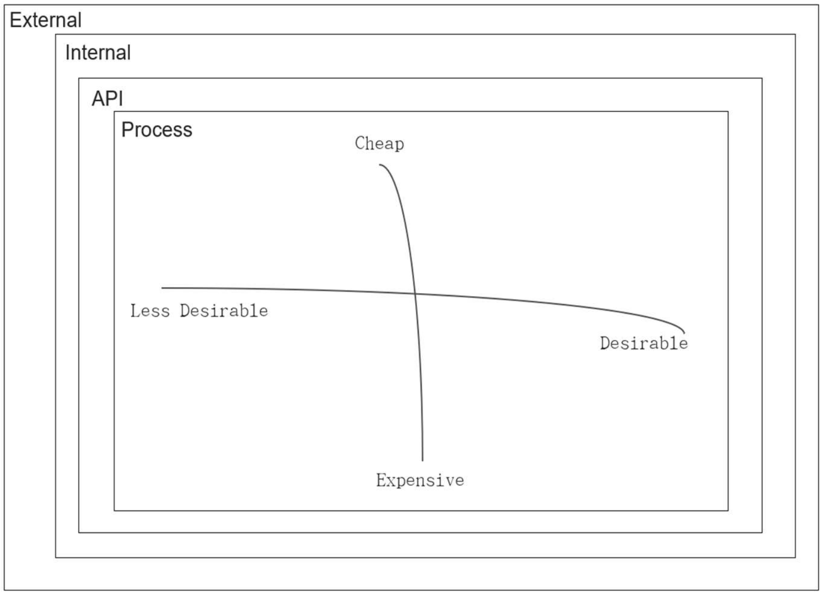


Figure 18. Prioritization Framework

4 Research Design

The research design section is split into three parts. First explaining the empirical part of the research and methods used. Thereafter, constructing the theoretical framework and describing the case company. Lastly, data collection and analysis methods are presented.

The purpose of the empirical part of the research is to identify APIs by using theoretical framework as a tool to be tested and improved. In the next paragraph an exploration about how the case studies are designed, what kind of methodologies are used in them in general and how evidence should be gathered and analyzed is presented. Case study as a research method was selected because of the boundaries of this study concentrating into a specific case company. Yazan (2015) compared the characteristics of case study, and states that Merriam (1998) sees "the case as a bounded system, thing, a single entity, a unit around which there are boundaries" and that is a lot more comprehensive list than Yin's (2002). Yin defines that case studies are preferred when the research questions includes "how" or "why" questions and when the investigator does not have control over surrounding events. Most of the case studies have been made in the field of politics, community psychology, organizational studies, regional planning and social science allowing the investigation to retain meaningful characteristics of a real-life events. Research questions in this study answers, "how to design APIs" and "how to prioritize APIs" which also indicated towards selecting case study as a research method.

This case study is exploratory in nature. The more data sources, the better it is for a case study, but there is always a risk getting lost in the data (Baxter & Jack, 2008). First, an understanding about the API field should be gathered, which is why a literature review part of this study is quite extensive. After having a solid understanding about API field, a valid and accurate mindset for setting up the questions for the interviews should occur. Every interview will be a part of a puzzle to determine, what kind of APIs are important in this field and which needs do they answer to. In this research data will be collected in a qualitative manner from limited number of key stakeholders by interviewing them.

4.1 Research method

The empirical part consists of a single case study. Case studies in general includes multiple steps to consider from defining the case to fitting design for the study that should be made. Research design includes at least research questions and propositions, analysis of the logic linking data to the given proposition and criteria for interpreting. Data for the research needs to be gathered as widely as possible, and the data should be analyzed. (Yazan, 2015) The quality of research can be addressed with validity and reliability, referring to construct validity, internal validity, external validity and reliability (Yin, 2002). Even though there is not a single correct way to report a case study, Baxter and Jack (2008) suggest that report should be a chronological story to follow and to gain trustworthiness few things should be looked deeper into. First, the research questions should be clear and comprehensible, propositions should be available depending on the research type, the design of the case study is appropriate for the research question, meaningful sampling and data collection should be implemented as well as correct analysis.

In case studies qualitative, quantitative or both can be used (Yin, 2002). This research was born from the case company's need to understand how they could utilize application programming interfaces in their business, which is why a qualitative approach was chosen for this research. since quantitative numeric data would not be able to answer this question. Because there were no experts related to API in the case organizations, it felt compulsory to collect all the relevant API researches together and give the case company vital information and knowledge as well as ability to evaluate and understand the research results thoroughly. Therefore, the first research question was formed as "How to design B2B APIs" giving guidelines towards design process and management of APIs in organization like the case company. Before any design can be made, it should be studied what APIs the case company should create first to gain as much value with least effort as possible. The second research question is formed as "what APIs offers the most value to the case company". The purpose of this question is to evaluate the needs of the case company and their customer to prioritize these API ideas and answer this research question. Because the question is sensitive to this specific case company, a case study method was chosen.

Furthermore, qualitative information about the case company's needs should be gathered to form API ideas from the business and customer needs. Interview study can be categorized into three groups: written interview, thematic interview or unstructured interview (Hirsijärvi & Hurme, 2000). For this research thematic interview was chosen for the method to gather such information around different themes. Method for constructing ideas from business needs and customer needs could not be found in the previous research, which is why it will be formed based on the themes. The interviews are extensive and might jump from topic to another, so there might also occur new unplanned questions and topic in the middle of the interview. There for it's a theme-based interview.

Interviews shall be guided by the template of questions including themes: customer, service and product itself, benefits and challenges of the service or a product for the customer and API. The first section of the questions is related to key stakeholders' responsibilities within services and products. The second section is about finding out the benefits and challenges in related services and products. The third section is about customer relationship; whether the key stakeholder is working with services or products that interact with external customers, what kind of needs the customer has and how is the case company unraveling the customer's need. The fourth sections are about API, whether the key stakeholder has experience or not, and how deep it is as well as suggestions and ideas about future APIs. The interview questions are presented in the appendix B.

After the interview all the audio records from the skype meetings will be transcribed and analyzed. All the needs and challenges will be gone through one by one to form API ideas. After all the API ideas are constructed from the interviews, an evaluation will be made. Because there was no such prioritization tool specifically for APIs, I modified prioritization tool for quality-based application requirements to fit APIs based on Otero et al. (2010). Finally, the outcome from this research should present a roadmap or a list of the APIs the case company should implement and in which order. The design principles should be used while starting to build APIs lifecycle management and the key performance metrics should be evaluated for each API to be analyzed and optimized later in the life cycle.

4.2 Case company description

The case company is one of the leading manufacturing companies in Finland with 12,500 employees worldwide. It operates in pulp, energy, paper and automation industries with more than three billion in net sales yearly. If the case company will be able to enhance the power of B2B APIs before its competitors, it might take its rightful place as a leader in the industry in the future as well. In this study, I only focus on the externally facing customer APIs which case company hasn't enforced yet in full force, and therefore only external components affecting business value are considered, even though external APIs should be exploited in internal staff services as well to fully understand their benefits and flaws (Benzell et al., 2017).

APIs create multiple possibilities for companies to expand their business. In manufacturing context majority, about 70% of the API calls are made in marketing or customer insights and analytics field, which have not been exploited by the case company yet. The share of internal API calls is only 10% in manufacturing context, which have been the most common use case so far in case company (Benzell et al., 2017). Therefore, it is recommended to explore what kind of opportunity lays ahead within customer facing APIs. Case company is in API discovery and experimentation stage, which is the first step towards API strategy, meaning that they are currently experimenting via PoC (Proof of Concept) and sandboxes to find out with limited integrations to core systems how to expose these to customers. Even though a clear API strategy is yet to be developed, they are already moving partially to the platform selection phase, where they have chosen Microsoft Azure as their API Management Platform. In this phase, organizations seek to prioritize core systems that are exposed via APIs seeking revenue growth by new products and services extending their business capabilities to wider audience (Holley et al, 2014). The next thing to accomplish would be completing the API strategy and defining the business needs that APIs answer.

4.3 Data collection & analysis

This study consists of two parts. The first part is the literature review and the second part is the empirical study. The literature review explains how APIs should be designed and managed, helping to understand the context of API in business environment explaining when and why it should be implemented. To perform the second part of this research such knowledge level of API is needed. In the second, empirical part of this research I interviewed 14 employees presented in the table 10. from analyst to vice President, from various fields in the case company to gain understanding about the business environment, customers and business needs. Interviews lasted from 20 minutes to almost an hour, approximately interviews lasted 35 minutes. All the interviews were littered and stored by the instructions of the case company. Because these interviews include a lot of information that might benefit competitors of the case company, the littered material is sealed and not published within this research as well as the results of the analysis, where API roadmap will be presented.

PEC #	Duration	Alias
1	25 minutes	H8DI
2	53 minutes	H10DI
3	50 minutes	H7SM
4	20 minutes	H6SA
5	25 minutes	H12VP
6	31 minutes	H1HO
7	45 minutes	H4DI
8	40 minutes	H5MA
9	37 minutes	H2SA
10	20 minutes	H13VP

11	30 minutes	H9AN
12	50 minutes	H3SM
13	40 minutes	H14DI
14	43 minutes	H11GM
15	30 minutes	H15SM

Table 10. Interviewees roles and interview durations

After transcribing the results, I analyzed the interviews to find out what kind of APIs case company should implement in the future by evaluating the results of interviews. Analysis started by identifying how interviewees described they're products and services, that brings value to customers. After making these notes into the transcribed material, I went it through again with "API glasses" in my eyes. Every challenge, proposal for development, renewal, technology change, update or idea they had for their service or product I considered as an API idea. I started to collect these API ideas into a table. The logical chain of reasoning for API ideas goes backwards, first I found out what needs to be done and then I started to categorize the idea and put the idea into the correct context extracting information step by step.

API ideas are short descriptions of what should be done aka a scenario. I extracted the idea into a description and requirements. In description it is described in more detail what kind of data or functionality should be performed into a system(s) whereas requirement states what tasks of functionalities the API should provide for the customer. Every requirement is performed for a reason and could be categorized based on ISO/IEC 9126-1 presented in section 2.9. Business value type is the higher level of this specific requirement, called business value type in this research. This value "type" is later used in the prioritization model. Business need can be found by analyzing the answer to question "how this API brings money back to case company?" If this question could not be answered, then a business need could not be identified, whereas key performance indicator (KPI) should be defined based on the value the API is supposed to inflict. Based on the KPI the use of API can be measured. Finally, the API ideas were prioritized based on the prioritization model and top10 APIs were collected into a list. Because the list does not provide a variety of information about why a specific API should be implemented I wanted to present them in a visual framework. It shows directly a visual roadmap, where should the next APIs be implemented in.

In this study case company acts as the registry administrator, taking responsibility from collecting and handling the material, therefore I will not be following data processing agreement of University of Jyväskylä (University of Jyväskylä, Data Processing Agreement) but instead follow case company's instructions. I will follow HTK practice (University of Jyväskylä, Privacy notice information) and case company's Employee privacy notice which states "Employee data will be retained as long as it is needed. Some data will be deleted at the time of termination of the employment. Some data will be maintained for a longer period because of the legitimate interest of the data controller, for example to maintain project information". Managing data practices shall be ethically respecting the case company's subjects' rights which includes information access, objection to processing, restriction of processing, data portability, data rectification and erasure. I will use privacy statement and form of consent if case company requires it, and I shall follow the policies of University of Jyväskylä if doing so (University of Jyväskylä, Form of Consent). Data is collected and recorded via individual skype interviews from November 2018 to January 2019. Data is stored in my personal archive. Interviewees were selected by my supervisor and Integration service manager responsible for APIs in the case company. They selected 9 interviewees together. Later, during the interviews I gained hints and tips who might have even more information and hidden knowledge, so I included 5 interviewees more to dive deeper into the case company's business and customer needs.

5 Empirical research results

The empirical research consists of three phases: API discovery, API analysis and API prioritization following systematically the research framework presented in chapter 3. This section describes and discusses the key findings of the empirical research, where two different kinds of research results are presented. Empirical conclusions (EC) are general findings made during the empirical research. Preliminary empirical conclusions (PEC) are the higher-level conclusions made based on ECs combining into more universal conclusions. Thus, each PEC is either a final conclusion of multiple EC's or independent preliminary conclusion.

5.1 API Exploration

KEY STAKEHOLDER

The process of selecting key stakeholders was executed by Integration Service Manager responsible for API development in case company and IT Services and Platforms Team Manager (case company experts) responsible for services and platforms in the case company. Skype meetings were arranged and suggestions about participants started flowing. Discussions were straight forward, since they both had strong vision about who has the correct knowledge in the case organization. They used their previous knowledge of the key stakeholders to estimate their API awareness, business knowledge and customer knowledge.

Observing the conversation about whether a key stakeholder had been involved previously with APIs or other new technologies for the case company, directly lead the case company experts to draw conclusions about specific key stakeholder being fit for the research. Nonetheless if a key stakeholder had been a pioneer and a leading example in his or hers career it led the case company experts to weight whether the key stakeholder was a fit or not. Customer knowledge was used more as an excluding factor if a key stakeholder did not work anywhere near the customer they would not be fit. Roles were also discussed briefly, and it was intended that as many roles as possible would be available for this research to gain wide image of the area in case company. Case company experts decided to start with 8 key stakeholders and if needed, we would search for more. Later, during the interviews it was discovered that a lot of the employees who had deeper knowledge were not known by the case company experts and thus I decided with them to include 5 more key stakeholders based on the information gained from the interviews. It would have been beneficial to have more versatile group of key stakeholders across the case company to get all-round representation of the case company's and customers' needs. Some of the business units were not incorporated in the study, which means there might still be hidden API ideas to explore.

EC 1 All relevant key stakeholders should be selected carefully

---

API AWARENESS

In this study I have broadened the key stakeholders to involve also other than salesman and account manager and the investigator is not an API expert as Moilanen et al. (2018) suggested which leads to a point, where neither the key stakeholder nor the investigator of API Ideas has the needed awareness to fulfill the process. Therefore, it seems that the level of API awareness of the key stakeholders should be considered further researchers as a part of evaluation of the key stakeholders. On the other hand, if the investigator is an API expert, this kind of evaluation is not needed, like Moilanen et al (2018) presents.

Key stakeholders’ API Awareness fluctuated considerably. Some of the key stakeholders did not know at all what API was.

“What is this API?” [H8DI]

Whereas some were not sure what it means today and avoided the question by stating having outdated experience of APIs.

“I have experience from the 90’s." [H12VP]

Few key stakeholders, like key stakeholder 4, had a clear vision of APIs and how they should be used in a strategic level.

"First we need to have an understanding what needs to be accomplished -- around APIs and data storages. -- In the end, these are the enablers that we need to support new innovations." [H4DI]

Yet most of the key stakeholders did not know how APIs could be used as a business tool to increase customer stickiness or net income.

"As far as I know I don't have experience, but I still might, if I have misunderstood it. Isn't it an interface which is built between systems to read each other?" [H1HO]

It was clear that the key stakeholders that had vision about APIs, also had presuppositions -specifically the word API economy provoked the key stakeholders in a negative manner as a trend -word. A lot of the knowledge the key stakeholders had was either old or limited to specific aspect of APIs. Key stakeholders' ability to understanding and experience what API means today influenced how well they could form API ideas independently. Nonetheless, if the key stakeholder wasn't sure what API was or their knowledge were not up to date, API ideas could be formed from the business needs or challenges met in the products or services the key stakeholders is responsible for.

Even though most of the key stakeholders did not have a clear understanding about what API means and how it could be used in the case company, ideas could be formed nonetheless. Thus, some of the ideas are not actually APIs in a technical manner, but instead applications, integrations or other services that could use APIs as a part of the implementation.

EC 2 Lacking API awareness influences the quality of API ideas by addressing broader concerns than only API related

---

ROLE AND BUSINESS KNOWLEDGE

Roles were clearly visible in the interviews in many ways implicating the overall business knowledge in the case company. Business knowledge indicated the level of history knowledge and wider visibility over the organizational units and regions whereas the role indicted the level of understanding. Also, the amount of time a key stakeholder was willing to spent varied as well as the personal interest or attitude into the study. The higher a key stakeholder is in organizations hierarchy, they saw the study as a way to improve processes in general whereas the lower the key stakeholders were in the organization hierarchy, the more they saw an opportunity to fix the issues they had in their own products and services. Overall the lower the hierarchy the more the key stakeholders were invested in a personal level, whereas higher roles just gave needed information. Apart from few irregularities this was seen in multiple occasions, which led the API ideas to have different level of ideas, from a single integration into a whole ecosystem. Roles like Vice President and Director emphasized top-level strategies in their work tasks, which inflicted that they needed to have a comprehensive understanding about the business in the case company overall. Roles like Solution Architect and Manager showed expertise in particular tasks they were involved with. The higher a key stakeholder were, in most cases the time for interview was more difficult to arrange and the key stakeholder might not be that creative since they might feel pushed or forced into it. The lower the role was in the hierarchy, the more the key stakeholders were giving plenty of time and they were more interested in the study in general.

"We are functioning at 35 countries and over 700 people are working in service tasks. My function is to lead this service organization. -- We are developing strategies -- products and services, that country organizations can utilize in their customer services." [H12VP]

"My function consists of employee-, subcontractor- and service management around the subcontractors in strategic level and the biggest time-consuming function is to communicate with business." [H4DI]

Looking from top to down in the hierarchy, the visibility diminishes, but instead deepens into a specific area of business areas. Solution Architect and Manager roles were involved in more specific functions, where they need a profound knowledge about a specific part of the business. This can be seen as a specific list of tasks and processes to be fulfilled, whereas the higher roles in the organization hierarchy included leading, communicating and otherwise organizing the internal group of workers to fulfill the business needs.

“I am responsible for application architecture -- what other solution can be found from IT world, their beneficial use and utilization.” [H2SA]

“*Capital Sales related to fabric, pricing, specking and all that comes with ordering process.” [H11GM]*

Yet the role of the key stakeholders is not a stable attribute, a key stakeholder can always apply for a different role in the organization. Every role has a specific purpose in a company, which might vary across organizations. Therefore, the role might not be all black and white solution for finding the best key stakeholders. Instead it could be used as an attribute to evaluate whether the key stakeholder has the level of knowledge demanded. This means that if a key stakeholder has a certain role, we can at least assume that the person has at least the level of knowledge needed to fulfill that role. Not many high-level roles are needed in order to find out API ideas. To construct specific API ideas, the valuable information is in the lower roles, who still has enough visibility across the organization. It was hard to evaluate the knowledge of a key stakeholder based just one interview and the limited amount of questions asked.

“The more diverse work experience employees that work in strategic guidance has -- the easier it is to work in developing environment. If the experience is deep and narrow, there is a long way to developing a comprehensive visibility.” [H4DI]

EC 3 The role of the key stakeholder indicates the scale of knowledge in the organization

---

CUSTOMER AWARENESS

Customer awareness is the ability to put oneself into customers shoes in order to form API ideas that would be beneficial to customers as well. However, some of the key stakeholders had difficulties separating the actual customer from internal “customer” that multiple functions are supporting internally. Overall the attitude towards customer might also affect the key stakeholder; whether the customer is a liability or an enabler. Even though that was not asked directly, some answers demonstrated the attitude as well.

Key stakeholders who did not work closely with customers but was involved in fulfilling their needs could easily form independent API ideas. Even though the customer is seen as a mass, the attitude towards them is positive and enthusiastic.

“It’s a mass somewhere, a grey unidentified mass which we try to serve as well as possible“ [H1HO]

Even though some key stakeholders thought that in their line of work customers are present every day, they did not see it as a compulsory part of their job to be in contact with them, which might indicate that the attitude is somewhat indifferent. This could be seen as a lack of ability to form API ideas. Since the key stakeholder did not have an actual connection with the customer, the key stakeholder was unable to see the benefits customer might gain from APIs. This indicates that it’s more important to try to understand the customer and their needs, than being involved with them.

“If one would not want to pay attention to customers, it’s not compulsory to be in contact with them. I am acquainted with the customers. They are present every day in my line of work.” [H12VP]

This perception is also supported with a second observation, where repeating conversations with the customers led the key stakeholders to be able to form multiple API ideas and give direct examples and problems they are facing. The more the key stakeholder is involved with the customer, the better the ability is to put oneself into customers shoes. Even though there is not a direct implication towards the attitude, it can be said that it’s not negative nor indicating any frustration, which could be the case when being involved with customers daily.

“We are constantly having conversations with customers about how things should go.” [H14DI]

Some key stakeholders were only involved with the customers if needed, in specific occasions. The customer is seen more as a liability than an enabler. This gives the impression that customer is not needed in every day work nor are their needs that important. In this case it’s hard to say whether is the lack of collaboration or the attitude that caused the API idea construction being impossible.

“I am not directly working with customers -- unless we are working on common development or project.” [H6SA]

EC 4 Customer awareness influences directly to the ability to understand customer’s needs

PEC 1 Role, customer and API awareness affect the quality of a key stakeholder in API exploration

API CANDIDATES

API Candidates are services or products, where APIs could potentially be used. In theory, APIs could be used in almost any context to make services more reliable, faster or functional. There is no such limit where it could be said that APIs do not bring any value into this. However, it is more about which context bring most value to make the extra effort to create APIs, is it worth it in the end or not. Every service or product offered to customers is potential API candidates. Because most of the key stakeholders had none or little understanding about APIs, I concentrated on gaining as much information as possible, from the products and services the key stakeholders were responsible for. For example, online catalogs were the most practical and easy API candidates to discover, since the user base exists already, and the technology being used currently is slow and requires a lot of manual work. The data is relevant to the customers who are interested in buying products and the easier and faster the catalogs are made to use, the better for the customer. As the following interview extracts demonstrate, valuable API candidates to consider are offering valuable data towards customers.

“We are selling technical fibers and textiles.” [H11GM]

“Electrical part sales where we have our supply presented as a catalog” [H6SA]

EC 5 API candidates to consider offer valuable data to customers

Some of the valuable content to customers includes functionalities and services that the customer can benefit. Offering solutions and expertise to customers is one of the key functionalities that manufacturing companies can offer. Therefore, building services that benefit the customers own performance, safety and predictability are in the core business in APIs perspective. All the API candidates are providing benefits to some parties in the business, but looking into the B2B field, the concentration should be in the cases where value is born to customers. The value is not always raw value that one company can bring to another, but more like giving the needed information, technology or expertise to increase customer’s efficiency, durability, safety or performance.

“We are giving guidance about how they should run their facilities and which direction it should be developed.” [H12VP]

“We can offer three kinds of applications to customers. -- How can we increase customers production line performance -- how to predict how long a device and its parts will endure as functional -- visibility about how much have they produced and consumed raw materials” [H7SM]

“Automation solutions that are based on a common hardware platform and all around is software products in different layers -- problems that we solve with simple basic software up to much more complex” [H10DI]

EC 6 API candidates to consider offer functionality that helps customers

PEC 2 API candidates to consider are online catalogs, products and services that offer functionality or data to customers

API OPPORTUNITIES

Every service and product have challenges. In our research model, we call them API opportunities. In some cases, the challenges were more in the pre-conditions that had to be done before actually using the service or product. At this point I searched for opportunities where APIs could be used.

“Creating a service request is not challenging, but finding the component is. Product names and codes changes rapidly” [H14DI]

Few other interviewees explained how data management is poor and it’s causing trouble and extra work later in the processes. Data does not always meet the requirements regarding data quality.

“Master data management, pricing, updating availability information and keeping it up to date needs resourcing“ [H8DI]

“It’s all depending on the fact that how the information is in order” [H11GM]

“Data quality is causing challenges, since it’s not always what we expect” [H14DI]

It has been seen challenging in the past also to keep the data up to date because the validation and maintenance of master data has been misplaced. The data should be harmonized, but the history of a big manufacturing company can create challenges regarding data harmonization.

“Validating, maintaining and organizing master data for customer and machine information has been historically an area where we have been significantly weak. Maybe because have done multiple corporate acquisitions that have brought legacy information and standards. Our problem is that the data is not harmonized everywhere.” [H13VP]

Some key stakeholders did not have a clue what should be done, but even those knew something needs to be done.

“How could the data be offered to customers and what is the cleverest way of doing that? That is the question mark.” [H2SA]

EC 7 Discussing API opportunities reveals concrete various challenges and pains in the API candidate

---

Key stakeholder 14 recognized automation needs towards customer. Even though the key stakeholder mentions integration, it is safe to assume that it means API -like integration where the same interface is published similarly to all customers - not just point to point integration. By these integrations both the case company and their customer can save time and resources, and at the same time the customer is being established by investing in the collaboration.

"Customer could pull data into their design systems with an automatic integration" [H14DI]

Also, the key stakeholder 2 were thinking that data should be combined more and reusability for the data that we have should be increased. Also, the key stakeholder thought even further: automation could help to manage data continuously by offering self-service to customers. Customer could update their information themselves, meaning that the data will be automatically up to date.

"Combine data more and apply the data that we already have. Also, we should offer more self-service to customers, so that they automatically keep the data up to date towards us." [H2SA]

Key stakeholder 11 combined both ideas suggesting that not only could we offer information to customers via integration, but also update information online towards customer about the current state of for example delivery times.

"We would be a lot more valuable for the customer -- If the customer could easily see from their own system what products they have, and we would share information, like delivery times with automatic updates" [H11GM]

It seems that most of the API opportunities, that could be improved in the current products and services offered to customers are related to automatically transferring information and to the fact that the information should be reusable.

EC 8 API Opportunities to consider includes reusability and automation needs

Even though every API candidate had challenges, not all of them could be solved directly with API. There were multiple situations where I could not form a challenge into an API opportunity, even though API might be part of the solution in the future. One key stakeholder had challenges in delivering drawings and unstable interface. API could be a better way of doing the same functionality, but more investigation about the problems in the interface needs to be done, before any conclusions can be made.

"There have been challenges in delivering drawing conversions. It's the worst if the interface breaks down, the offer or it's update won't go through." [H3SM]

It was also difficult to make the key stakeholders to understand what kind of challenges could be solved with APIs and which ones are just not relevant at all. Questions asked did not restrict the answers in any way, so the answers vary a lot depending on how they comprehended the question. But in the end, if the question was narrower some challenges might not come up at all since the API awareness was limited among key stakeholders. If the question would have been API specific challenges with customer base might not have come up.

"The challenge is the heterogeneous customer base. How can we offer targeted and relevant information to those customers?" [H1HO]

"Our product has a long lifecycle, some of these deliveries are far from the past. Electrical systems did not exist at the time, which makes it challenging to produce to everyone in the same manner." [H6SA]

EC 9 API opportunities can be revealed by gaining knowledge of API candidates challenges, but not all challenges can be formed into API opportunities

API VALUE PROPOSITION

API value proposition is the added value that API could bring to existing product or service. Because a perfect product or service do not exist there is always something you could do more securely, faster or efficiently. API value propositions were found from extremely high demand services and products offered to customers. In this kind of critical benefits to customers could be created via APIs, even though, often there is a lot more needed.

“A lot of value-add is created in performance optimization -- Weather it's consumption, energy usage or minimizing the maintenance breaks.” [H14DI]

API value proposition can be determined after gaining knowledge about how the product or service brings value to the customer in the first place. Only after that can the API value adding features be formed. Key stakeholders were eager to share API value propositions, where as many of these wishes weren’t APIs technically but more high-level requirements. Nonetheless, the value of implementing the API might exactly meet the propositions characteristics or functionality.

“Increasing the customer’s processes capacity and quality or efficiency.” [H13VP]

Some stakeholders couldn’t really tell where to use APIs in their service or product, but they were able to see the benefits.

"If multiple systems exist to implement a process it would be more efficient to use only one interface from end to end." [H4DI]

Few key stakeholders were able to form independent API ideas already in this phase without asking that specifically.

“Clever chatbot telling our customers where to find information.” [H1HO]

EC 10 To understand how API brings value, the added value to customer needs to be understood

---

BUSINESS NEED

Business needs were hidden in the interview, but they could be extracted from there by identifying adjectives and combining them to corresponding ideas the employee presented in the interview. Thus, most of the identified business needs were not referred to any idea but instead by deeply analyzing them, they can be linked to API ideas as requirements. Not all business needs were clearly requirements, but more like wanted outcomes or even API ideas. For example, key stakeholder 11 was able to describe the need for visibility as an outcome, what the API should provide to customers, which are requirements for APIs. Whereas the description what data is needed, aka what machines customer has is actually an API idea.

"The daily face for customers should have as good visibility as possible to that customer; what machines the customer has and what's going on at the moment in each one." [H11GM]

Key stakeholder 6 nonetheless mentioned business needs that were purely requirements. These requirements are recognized by ISO/IEC 9126-1 standard for Software quality.

"Online shop needs to be available and running in order to find the needed products in a secure way." [H6SA]

Thus, key stakeholder 2 and 11 had a clear API related business need about data quality and user experience in the interface.

"The goal is that we have up to date information about ongoing transactions and state of the company functionality" [H2SA]

"It should be the goal that the customer has only one interface with hundreds of programs underneath." [H11GM]

EC 11 Business need describes wanted outcomes and requirements for API ideas

CUSTOMER'S NEED

Customers were interested in getting basic information about themselves, like history of purchased products, current machinery and other related information that is vital, for the customer to manage and run their business. It's not specifically creating new value, but if this information is missing or the process to get this kind of fundamental information is too slow and difficult, it's decreasing the efficiency for both of the companies; customer and the case company.

"When customers are asking for specific information, you cannot find it anywhere. What did the customer buy some years ago? -- It is a lot of manual work to find those things. I think in the future the customer will be more interested to have information about their products they have received from us." [H10DI]

"Customer is probably interested in what machinery they have, and we should be able to provide this information to them. What have they bought from us and what they currently have in use" [H2SA]

Customers would get more value from data, that would have been processed; filtered or enriched. The case company has multiple data sources and history data from their own products and processes, which case company should start to use in order to gain value - not only to the customer, but also to themselves. By combining data sources and offering more self-service to customers, the customer's needs would be better satisfied.

"Combining data from multiple sources and transforming the data we already have and providing more self-service" [H2SA]

"Storage accounts should be available through API to calculate and show the prices around the world." [H3SM]

Not all key stakeholders were able to specify what data is actually needed, but still saw the potential value.

“Providing unique data to market would benefit customers investment planning” [H5MA]

Some of the employees did not have any contact with the customer, which made it harder for them to understand the overall process from customers' point of view. Furthermore, it made it impossible for some of the key stakeholders to define customer needs, only internal business needs. Even if the key stakeholder had an understanding about the customer in general and what kind of services they use, it was not clear what kind of value customer might get from API.

EC 12 Customers needs can reveal API ideas

PEC 3 API ideas can be discovered by interviewing key stakeholders about inflicted value and challenges in their product or service

---

5.2 API Idea Analysis

API IDEA

API idea is the constructed idea from the key stakeholder from the previous API exploration phase. After listing all the API ideas, there were 32 ideas in total from 11 key stakeholders, whereas 4 key stakeholders did not provide any API ideas not could ideas be formed from the interviews. I analyzed API ideas one by one, trying to find the business need, if one did not already come up in the interviews. During interviews I had gained a miscellaneous information about API ideas, business needs and customer needs. I combined them in a table, where it would be easier to compare the data. Using Dayley & Oliffe’s (2017) categorization about business needs - why APIs should be implemented in the first place, made it easier to understand the higher level of APIs purpose. If the business need could not be identified, it led the API idea not being complete and therefore could not be further analyzed. This could indicate that either the questions asked were not comprehensive enough, or that the key stakeholder just was not able to give all the needed information. This stage of analysis already cropped out one API idea from key stakeholder

9, because I could not figure out how and where this would be used, and the key stakeholder did not cover this either. The idea is too general without a specific context.

"Analysis API which compares internal and external analysis information" [H9AN]

The categories for the business needs were: enable web and mobile applications, integrate internal applications, interface with microservices, publish data, create Cloud and SaaS integrations, enable IoT interactions, engage customers and extend business, which could use to all other API ideas. Because it's easy to put an idea into one of these categories, this is not a dividing line between implementable APIs and those which can't be implemented. For example, key stakeholder 1 had an idea about learning chatbot as well as key stakeholder 11. Furthermore, a total of 5 key stakeholders had similar core idea about Installed base API, where fundamental information would be offered to customers directly with abilities to view, search or order. Overall the ideas that had multiple key stakeholders behind them, were ranked higher in the top 10 list of API ideas based on desirability presented in section 5.3.7. The higher the desirability is, the higher priority the API has in the case company. In the table 11. below are examples from API ideas that had different business needs behind them and for further analyzation also desirability has been added to the table.

API Idea	Business need	Key stakeholder	Desirability
Chatbot API to connect customers to correct contact person to help with a problem	Enable web and mobile applications and IoT interactions	H11GM	87%
Learning chatbot combining multiple source's information	Enable IoT interactions	H1HO	60%
Installed base API; machinery and related documents	Publish data, engage customers, integrate internal applications	H2SA	78%

Installed base API; machinery, search and order products	Publish data and engage customers, integrate internal applications	H8DI	75%
Collaboration API; view purchased products and related documents	Publish data and engage customers	H10DI	64%
Product document API; view product related documents	Publish data and engage customers	H11SM	64%
Installed base API; product hierarchy, view, search and order products	Publish data, interface with microservice, engage customers	H14DI	75%

Table 11. Business needs for API Idea

PEC 4 API ideas that have multiple key stakeholders behind them are likely to be more beneficial and therefore have higher desirability

SCENARIO

Scenario is an explanation of the needed functionality of an API, because a more specific explanation of what functionality the actual API would provide was deemed necessary. Therefore, scenario was the first thing to consider during analysis explaining the use case of the API idea. Despite that, to expand the API idea into scenario, a lot of case company specific knowledge was needed. I had already been working 2 years in the case company, which made it possible to provide such information. If the analyst did not have this sort of case company history knowledge, there would be need for other sources of such information to understand the case company’s services, processes and products. In some cases, the analysis revealed a lot more information that was given by the key stakeholders and in some cases new information was not found. Example of how the scenario could be formed into a scenario in table 12.

API Idea	Scenario	Key stake-holder
Transaction API about current transactions and their states	Provide reshaped up to date information about company transactions and states	H2SA
Show more personalized and relevant data	Provide filtered data based on CRM and company business line to related systems	H1HO

Table 12. Scenario for API Idea

EC 13 Scenarios could be formed by combining the API idea with case company knowledge

REQUIREMENTS

Requirements describes the needed functionality, the actions the user can perform or expect from the API. The process for defining at least the basic requirements for each API was a laborious task, and still I cannot say that it's complete. As soon as I noticed how much work the analysis of this is, I decided to just define the MVP, which is the least amount of functionality that needs to be implemented in fully functional API. It would have been a lot easier, if the key stakeholder had the ability to give this information in the interviews, but because of the poor level of API awareness, this was not possible in many cases. Therefore, my personal level of knowledge affected the requirements a lot. Depending whether, I had a sufficient understanding about services or products in question, I might have missed some MVP related functionalities. One could take endlessly time and invent new requirements over time, but also because of the limited time to finish this research I had to limit the time used in this part of analysis. I was able to extend the API idea from key stakeholder 2 as showed in table 13. because I was familiar with the API candidate itself and therefore able to form requirements. Some of the requirements came up in the interviews but were indeed hidden behind different sub-categories.

API Idea H2SA	Requirement
enables customer to manage orders and view related documents	Create, confirm and view order
	Track shipment states
	view history information: see what has shipped to customer and what has been ordered
	what and where is the item in hierarchy

Table 13. Requirements for API Idea

EC 14 API Requirements were time consuming to define

BUSINESS VALUE

Business value describes what kind of value the API is inflicting. To define the business value, it needs to be analyzed what the actual API would affect in real life, once it's at use internally and externally. Similarly, like defining the requirements, not all values might be revealed here, but the ones that are visible from the information available. If further research is being made towards these API ideas, more business values might occur. The analysis is based on the requirements defined, where it is described what kind of functionality is needed from the API. Business values found were then categorized into the quality model for Web API presented in section 2.6. Each requirement should provide value and therefore looking the API idea "*Installed base API with machinery information, search and order parts ability*" from key stakeholder 7, following business values were determined based on analysis requirements below in table 14.

Requirement	Business value
Customer can view own machinery	Increase Transparency
Customer can order spare parts without manual work	Increase speed of process
Search ability to find spare parts	Increase quality of service and speed of process

Table 14. Requirement analysis to form business value

EC 15 The business value of an API can be defined based on requirement

PEC 5 API analysis helps to define business value for the API idea

5.3 API Prioritization

SCOPE

Scope describes how big of a user pool the API would have. Internal gives 33%, external 66% and both 99% desirability like presented in table 19, which will affect overall desirability by 15%. Because previously analyzed requirements describe the needed MVP functionality, the scope was straightforward to define. If the use cases included only internal information and processes being automated, it was analyzed to be internal API, even though the customer still gets the benefits. However, if the API will not produce new functionality nor benefit internally, but towards customers instead, it was analyzed external. Lastly, if API could benefit both the case company and the customer, it was analyzed to be both; internal and external API. Example of how the scope was determined from requirements and the desirability percent gained presented in table 15.

Requirement	Scope	Desirability	Key stakeholder
Internal application can create mobile applications easily and fast in a standard way	Internal	33%	H2SA
Project information integrated directly to customer's project management system	External	66%	H14DI
Customers and employees can find needed person faster	Internal & External	99%	H15SM

Table 15. Scope from the API requirement

EC 16 The scope of an API can be defined based on the requirements

TYPE

API ideas were categorized into 3 types; functionality, improvement or a product. If an API created a new minor functionality to already existing system or otherwise it could not be defined as a complete improvement to existing system or product, it was analyzed as a functionality. On the other hand, if an API gave huge new features, that changes most of the characteristics of a product or a service, it was analyzed as an improvement. As well as, if the API improves the product or a service overall. Only if the API changes the whole operating logic of an existing product or a service, could it be analyzed as a product, otherwise APIs were analyzed as products whenever they were completely new offerings to the case company. The type was not always a simple definition, but more like a resilient conclusion at the time depending on my personal case company knowledge and the scenario enriched from API idea like presented in table 16. Key stakeholder 3 suggested a simple automation functionality to existing Supply chain system, whereas key stakeholder 4 suggests a wide range of functionalities to existing Project collaboration system, that will increase the quality of an existing system in full. In order an API to be analyzed as a product, it needs to have something unseen. Like the key stakeholder 5 suggested a warehousing API, with completely new abilities that would help collaboration, increase automation and increase the quality of service overall.

Requirements	Type	Desirability	Key stakeholder
Automated purchase order sending	Functionality	33%	H3SM
Better visibility to project information and related documents. All information in the same place and no need to switch between applications to get all information.	Improvement	66%	H4DI
Create order suggestions automatically, ability to estimate need for products ahead, enable customers to see what case company have in stock and vice versa	Product	99%	H5MA

Table 16. Type from the requirement

EC 17 The type of an API can be defined based on requirements

---

API SPECIFIC

API specific attributes can be defined from the business need of an API. Each API could have more than one API specific attribute fulfilled, which means that each attributed being realized adds the overall API specific desirability. The target for API specific is 70%, which means to gain full 100% percent desirability 5 out of 8 should come true. If an API idea has for example 1 of these API specific attributes fulfilled, the desirability would be 18%. Because business needs are categorized based on ISO/IEC 9126-1 standard, it was easy to create the link between the API specific attributes and business needs categorizing the business needs under ISO/IEC standard, example in the table 17. below. However, sometimes the same business need could be categorized under different API specific attributes if the requirement of the API described a different function. For example, business need “increase service quality” can be categorized as efficiency: resource utilization or functionality: interoperability. Therefore, the requirement cannot be left out completely. Requirement should be used whenever necessary to define the correct API specific attribute the business need belongs to.

Business need	Requirement	API Specific ISO/IEC 9126-1	Desirability
Increase standardization	Machinery have a hierarchy defined	Usability: Learnability	18%
Increase functionality	Customer can view machinery, history of purchase orders and end of life dates	Usability: Operability	18%
Increase self-service	Customer keeps their information up to date	Reliability: Maturity	18%
Decrease email phishing	No more sensitive data sent via email	Functionality: Security	18%
Increase service quality	More variety of devices allowed	Functionality: Interoperability	18%
N/A	N/A	Functionality: Compliance	18%
Increase speed of process	Customer does not have to find information manually	Efficiency: Time behaviour	18%
Increase transparency	View related transactions and states at the same time	Efficiency: Resource Utilization	18%

Table 17. Business needs from the requirement

EC 18 API attributes can be defined based on business needs and requirements

PMF IMPACT

PMF impact should be evaluated by an expert of the case company, but in this case, I acted as such completing the prioritization. I analyzed whether the API idea, requirements or business need revealed information about the benefits towards different fields or not. Like API specific attributes, these stack one after another with a target of 0.9% and the maximum still being 99%. The weight is justified in a matter of inexperience; if I can only see the API relevant in one segment, the margin of error will be smaller. If the API is perceived as a major functionality from either software, hardware of system side, it will gain 37% percent desirability. Whereas two of them gain 74% percent desirability. Overall impact was most difficult attribute to evaluate, since I am not an expert on either of those fields and the only way to define the impact is to answer the question "is this API perceived as a major functionality?". Example of what kind of API ideas gained each PMF impact attributes shown below in table 18 and the desirability percent gained from PMF.

API idea	PMF Impact	Desirability	Key stakeholder
Project information API	Software Lead	37%	H4DI
Chatbot API to connect customer to correct contact	Software Lead and System Lead	74%	H11GM
Installed base API with machinery information, search and order parts ability	Hardware Lead, System Lead and Software Lead	99%	H8DI

Table 18. Impact from the API idea

EC 19 Impact cannot be defined reliably without case company expert

CUSTOMER SATISFACTION

Customer satisfaction implies into the number of customers that would be satisfied from the API and would be able to use it to their benefit. Because in this study there weren't actual customers involved I acted as an expert and tried to put myself into the customer's shoes. I thought all the aspects from the customer's point of view and from the business point of view. Not all APIs would benefit smaller customer's but would instead benefit only big companies. Smaller customers might not have the needed resources to use APIs at all unless it's offered as a service. On the business side I analyzed whether the API offers beneficial data, service or other beneficial functionalities toward just one or few sections of the case companies reach, example in the table 19. Is the API for example restricted only to specific country, business line or area? Not one API idea were found, where it could be confirmed that only one specific customer could benefit from the API. This might mean that API ideas in principle were thought to be provided for larger group of customers, and these kinds of ideas did not come to mind in the interviews. Furthermore, if the API idea was an API providing information to customers that they should further analyse or process, it's not beneficial to all customers because of their limited resources. If the API was usable to all customers giving beneficial information, it was given the 99% percent desirability.

API idea	Customer satisfaction	Desirability	Key stakeholder
N/A	One	33%	N/A
Reference data API offering process data as a reference data	Few	66%	H4DI
Inventory API offering visibility towards inventories	Many	99%	H5MA

Table 19. Customer satisfaction from the API idea

EC 20 Customer satisfaction cannot be defined reliably without encountering with customer

---

PENALTIES

Penalties to be discovered from each API idea were costly, risky and complex. Not a single API can be implemented for free, so the line between normal costs and being "costly" was a difficult attribute to estimate. Otherwise complexity and riskiness were easier to estimate based on the case company knowledge. If the API included functionality, that was not trivial to me, I analyzed it as complex, examples how penalties were used in table 20. Risky could mean that API even if implemented might not pay back the investment placed into it or there were some other fundamental blockers to see the API thrive, at least yet. For example, in the table 20 the search engine API by key stakeholder 11, is a great idea for internal use, but to implement it would take a huge effort and it would not give any benefits to customers. Artificial intelligence will probably solve this issue anyhow soon, so the bicycle should not be reinvented which is why I analyzed it as risky. Also, when looking into penalties, the benefits of the API should be considered. In the installed base API from key stakeholder 14, the benefits are directly related to customers and the quality of service. So even that the API is complex and costly to make because of the master data's difficult reusability, it still should be implemented. Therefore, penalties can only be analyzed by a case company expert with a knowledge of such use cases.

API idea	Penalties	Desirability	Key stakeholder
Transaction API	-	99%	H2SA
Inventory API	Costly	66%	H5MA
Installed Base API	Complex and costly	33%	H14DI
Search engine API	Complex, costly and risky	0.01%	H11GM

Table 20. Penalties from the API idea

EC 21 Penalties cannot be defined reliably without a case company expert

DESIRABILITY

Overall 34 API ideas was formed and 32 of them could be analyzed to get desirability percent. After calculating the desirability for each API idea, it seemed that most of the API ideas got more than 60% desirability. Forming up the top10 list of most desirable API ideas showed that the 75% desirability is the limit to hit the top 10. The highest desirability was 87% percent of the chatbot API by key stakeholder 11 and the lowest desirability was 43% percent by key stakeholder 2, whereas the average desirability for API ideas was 71% percent. Based on the average desirability, overall the API ideas were good quality. In the below table 21. is presented all API ideas and how their desirability percent was constructed and the explanations for the abbreviations used in the table.

Type: F - Functionality, I - Improvement, P - Product
Scope: I - Internal, E - External, B - Both
Custom: O - One, F - Few, M - Many
API Specific: O - Operability, L - Learnability, M - Maturity, S -Security, I - Interoperability, C - Compliance, T - Time behaviour, R - Resource utilization
PMF: S - Software, H - Hardware, S - System
Penalties: C - Costly, R - Risky, C – Complex

API	Type			Scope			Customer			API Specific								PMF			Penalties			Desirability
	F	I	P	I	E	B	O	F	M	O	L	M	S	I	C	T	R	S	H	S	C	R	C	
1. Chatbot	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	87%
2. Scheduler	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	1	1	1	0	1	0	0	1	85%

3. Supplier Salesforce	1	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	0	1	81%
4. Transactions and states	1	0	0	1	1	1	1	1	1	0	1	0	0	1	0	0	1	1	1	1	0	0	0	80%
5. Warehousing	1	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	0	1	1	0	1	79%
6. Inventory	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	79%
7. Vendor measurement	1	1	1	1	1	1	1	1	1	1	0	0	1	0	0	1	1	0	1	1	0	1	1	76%
8. Installed base	1	0	0	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	1	0	1	76%
9. Contact	1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	76%
10. Project contract	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	76%
11. Project integration	1	0	0	1	1	0	1	1	1	1	0	0	0	1	0	1	1	1	1	1	0	0	0	75%
12. Installed base	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1	0	1	75%
13. Search engine	1	1	1	1	0	0	1	1	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1	75%
14. Installed base	1	1	0	1	1	1	1	1	1	1	0	0	0	1	0	1	0	1	1	1	1	0	1	75%
15..Report	1	1	0	1	1	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	1	0	0	73%
16. Customer chatbot API	1	1	1	1	1	0	1	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0	1	72%
17. Commenting	1	0	0	1	1	0	1	1	1	0	1	0	0	1	1	0	0	1	0	1	0	0	0	71%
18. Product	1	0	0	1	1	0	1	1	0	1	0	0	0	0	1	0	1	1	1	1	0	0	0	70%
19. Enriched data	1	1	0	1	1	0	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0	0	0	70%
20. ERP integration	1	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	0	1	1	0	0	0	0	68%

21. ERP integration	1	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1	0	1	1	0	0	0	0	68%
22. Update information	1	1	0	1	1	0	1	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	68%
23. Product	1	0	0	1	1	0	1	1	1	0	1	0	0	0	0	1	1	1	1	1	1	0	1	64%
24. Collaboration	1	0	0	1	1	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	1	0	1	64%
25. Personalized data	1	0	0	1	1	1	1	1	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	61%
26. Learning chatbot	1	0	0	1	1	1	1	1	0	1	0	0	0	0	0	1	1	1	0	1	1	0	0	60%
27. Request changes	1	0	0	1	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	56%
28. Analysis	1	1	1	1	0	0	1	1	1	1	0	1	0	0	0	0	0	1	0	0	0	1	0	56%
29. Documentation translation	1	1	0	1	0	0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	53%
30. Process data	1	0	0	1	1	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	51%
31. Process data	1	0	0	1	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1	1	0	0	48%
32. Standard API service	1	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	1	1	0	43%

Table 21. Binary Inputs to desirability

EC 22 Desirability can be used to prioritize APIs based on their requirements

---

PRIORITIZATION

The first APIs that should be implemented needs to be easy to complete but still provide a lot of functionality. API prioritization can be made just based on the desirability, but in the case company the focus was to find APIs that are most beneficial with moderate costs.

Therefore, I decided to further analyze the APIs within costs and complexity following the statement of key stakeholder 2.

"We have extremely easy APIs, but when we go towards bigger APIs, it might turn into more complex and wider entirety." [H2SA]

In the figure below the model is presented, costs in the horizontal axel and complexity in the vertical axel. In the left top corner are the API ideas that are complicated and costly, so they are not the priority for the case company based on their targets. In the top right corner there would be API ideas that are complicated but cheap, but the combination is ineligible to begin with, because an API being complicated will impact its costs increasingly. Therefore, APIs that are complicated and cheap cannot exists. In the bottom left corner are the API ideas that are expensive but easy to implement. This denotes that these APIs data or some other functionality is hard to create, but once that is available the API itself is easy to implement and reuse the content in the future. Lastly, in the right bottom corner are the most beneficial APIs to create. These APIs reading from right to left and bottom to up; cheap and relatively easy to make without any known obstructions.

Each API idea analysis is based on the history knowledge of the case company expert, which I was acting as in this study. API idea 12, which is the most difficult and expensive to make was easy to extrapolate; to get a "google-like" search engine, all the information in the case company should be available or made so. That is, first a huge work and because the long history and unharmonized processes, databases and services, I in fact know that this is somewhat impossible at this point. Whereas API idea 3, which enables customers to manage orders and view related documents is entirely possible. There is no technical limitations nor is the data too difficult to get. In fact, now some sort of actions has been taken in the case company, to a direction which would provide some information towards this API.

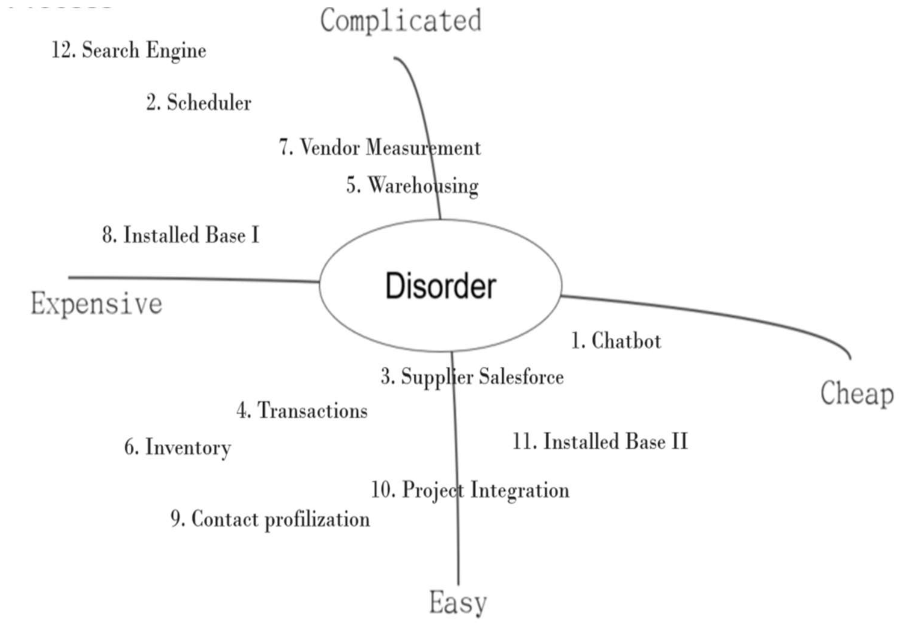


Figure 19. API Prioritization based on case company expert

API ideas were scattered evenly across. Result of the desirability analysis reveal that API ideas that where high on the desirability percent might not be the first ones to implement, when giving more weight to costs and difficulty of the implementation. Looking API implementation in the case company's perspective, APIs that are most viable are the ones that will give most benefits against costs. Therefore, it is surprising that for example Installed Base II API idea, which was the 11th on the desirability list, would be the second API to implement based on desirability analysis as well as 2nd Scheduler API, which should be implemented secondly is in fact second to last. On the other hand, many of the APIs kept their desirability level after desirability analysis, for example APIs 1, 3, 4, 6, 8 and 12. It can be inferred that depending on what the company weight most, it can change the order of the API ideas up to 50%, since half of the results were changed in the top10 API ideas. Overall the value gained from a desirability evaluation is significant. Desirability can still be used individually, but the risk of not giving enough weight toward correct metrics might affect the priorities.

EC 23 API Desirability can be further prioritized to fit case company's requirements

PEC 6 Desirability can be used to prioritize APIs based on their requirements and further analyzed based on case company's weights with case company expert analyzing the desirability

5.4 Summary of PECs

PEC 1 Role, customer and API awareness affect the quality of a key stakeholder in API discovery

PEC 2 API candidates to consider are online catalogs, products and services that offer functionality or data to customers

PEC 3 API ideas can be discovered by interviewing key stakeholders about inflicted value and challenges in their product or service

PEC 4 API ideas that has multiple key stakeholders behind them are more probably beneficial and therefore have higher desirability

PEC 5 API analysis helps to define business value for the API idea

PEC 6 Desirability can be used to prioritize APIs based on their requirements and further analyzed based on case company's weights with case company expert analyzing the desirability

6 Discussion

The discussion links the results found to the theoretical foundation and earlier literature by sifting through the primary empirical conclusions. The primary empirical conclusions and existing research provides practical implications. By deciphering on how to design, reveal and prioritize API ideas successfully in large manufacturing companies. Foundation is formed based on the established knowledge analyzed and the justifications that explains theoretical contributions of the research.

6.1 Research Implications

PEC	Description	Previous research material
1	Role, customer and API awareness affect the quality of a key stakeholder in API discovery	New
2	API candidates to consider are online catalogs, products and services that offer functionality or data to customers	Supporting
3	API ideas can be discovered by interviewing key stakeholders about inflicted value and challenges in their product or service	Supporting
4	API ideas that has multiple key stakeholders behind them are more probably beneficial and therefore have higher desirability	Supporting
5	API analysis helps to define business value for the API idea	New
6	Desirability can be used to prioritize APIs based on their re-	New

	quirements and further analyzed based on case company's weights with case company expert analyzing the desirability	

Table 22. Previous research material supporting results

SELECTING KEY STAKEHOLDERS

To find the valuable key stakeholders there are multiple factors to consider based on PEC 1. Like the first preliminary conclusion reveals, there are some attributes that matter more than others. It seemed that the role of the key stakeholder offers relevant information about the visibility the key stakeholder has as well as the awareness of APIs and customers. Meaning that, when evaluating and selecting key stakeholders for the API teams source of information, it should be considered whether the key stakeholder has enough visibility towards customer and if the key stakeholder is aware of the possible API benefits and able to identify customers key characteristics, needs and experiences (Barnes et al., 2018; Moilanen et al., 2018). Furthermore, the role of the key stakeholder should be considered on a case by case basis depending on how broad or contextual APIs are in the case company's target. In the case of the target APIs are crossing business and country lines, it should be decided whether to include directors and vice presidents of certain business lines or are the target APIs more specific to certain application, service or product. In this case, service managers, specialists and other lower level roles should be selected to gain detailed information.

API CANDIDATES TO CONSIDER

API candidates that should be considered offer valuable data and functionality to customers, like PEC 2 states supported by existing research material. Based on these existing catalogs and other online store -like services should be implemented first as an API towards customers. This does not only provide new ways for customers to connect, but also offers personalization and self-service. Customers can use the data and plug it in their own systems, which results in stickiness (Moilanen et al., 2018). Moreover, case company should start investigating and verifying which other services, products and applications are providing such valuable assets towards customers. Whether it’s a functionality, data or a service, it needs to be confirmed by the case company experts. This could be made by further interviewing internal company experts or by running a query, audit or simple targeted interviews to the biggest customers.

GATHERING API IDEAS

When companies don't have a clear business objective, it needs to be built. The building blocks are customer's needs, limitations and business needs and the requirements formed based on those (Alnabhan et al., 2014). Such information is not easily possessed by a single person; therefore, the information needs to be gathered from the key stakeholders like PEC 3 suggests that the API ideas can be formed based on interviews. Even though some of the ideas were not technically APIs but more like solutions, services or products that can use APIs, nonetheless valuable information related to APIs. Whilst this is acceptable, it should be noted that before any implementation actions are taken it is needed to filter these ideas further taking a closer look at the idea; is it an API or something else? Whether or not it is an API, is somewhat irrelevant, since APIs are part of the complete service or product as a technical enabler (Moilanen et al., 2018) Therefore it's important that open discussion and events to share ideas should be arranged more in general. Business knowledge should be utilized in an effective way, whereas currently a lot of potential knowledge is being dismantled and not used, because there is no hub for development nor inventions by all the stakeholders. If such a hub would exist the key stakeholder could change ideas and get support for their ideas, which would be beneficial regarding PEC 4; the more support an API idea is, the more certain it is that it has actual value.

API ANALYSIS

Business value is one of the first things to consider when weighing whether to implement and API idea or not. Because the concept of the API is not familiar to all key stakeholders, the business value of an API needed to be extracted from the interviews. The process of the analysis from the actual API idea, to scenario and furthermore requirements and finally to business value is a heavy process, that should probably not be performed in as structured way, but like the PEC 5 states it was indeed a helpful tool in a case where the business value could not be directly demonstrated. Instead the business value should be arisen from the business key stakeholders and if such business value is not raised and verified the API idea should not be further discussed until the needed attributes can be defined.

API PRIORITIZATION

PEC 6 suggests that APIs can be prioritized based on Otero et al. (2010) binary input evaluation method producing the desirability percent presented in the literature. The attributes that needs to be analyzed by case company expert are impact to the case company, customer satisfaction and risks aka penalties. Analyzing impact means to consider whether the API would have impact to other areas in the case company. For example, if the API reduces the need for hardware capacity it will have a high impact on hardware of the case company. Secondly case company expert is needed to analyze customer satisfaction; if the API would benefit most of the customers or only a certain group of customers, it is necessary to have expertise of the case company in order to create such assumptions. Lastly penalties, meaning the effects and possible risks in the implementation of the API is knowledge that only a case company expert can evaluate. Risks can occur in fetching the data being too slow or difficult as well as the complexity of legacy systems can cause the API implementation being nearly impossible increasing the risks. Other attributes to evaluate can be analyzed based on the requirements and API idea directly; scope, API specific attributes and type. Whether the API is intended to internal, customer or both, should be defined already, meaning that basically any analysis is not needed as well as the API specific attributes are directly linked to the requirements of the API and therefore analysis is already done. Thus, the type of the API still needs to be defined based on the invention level; for example, if the API is a new feature for existing product, it’s a feature, but if it’s actually improving the existing products functionality it would be an improvement. These attributes need to be analyzed to calculate the desirability percent. Even more accurate results can be produced when the API ideas are further analyzed by the case company expert and the desired weights are given to attributes that have critical impact in practice.

WHICH APIS TO IMPLEMENT

There multiple API types, like discussed by Moilanen et al. (2018), some APIs are directly in contact with master data, some APIs are orchestrating which APIs to call and how the information is formed, and some APIs are interfaces towards customer hiding the functionality. Depending on the maturity level of the case company, the prioritization list of APIs should be analyzed, like PEC 6 suggest. It should be further investigate which service level APIs needs to be existing before the experience APIs that most of the key stakeholders have been suggesting, like the PEC 4 states that the more key stakeholder stand behind the API idea, the more probable it is to have higher desirability. Prioritization based on the desirability percent should be further analyzed by API team, in the same manner that a new API should be analyzed. Before this can be done, the API team with API manager should be announced. The API team should be able to evaluate which data is most likely to have reusability needs in the future. APIs that offer fundamental information to other APIs and are easily provided should be the first ones to implement.

The most desirable API based on the empirical research and an in-depth analysis is the Chatbot API, which would combine the customer towards the correct contact person in case company based on profilization. Even though data service APIs provides filtered and unified interfaces, the RESTful data services use underlying data schemas which do not support continuous exploration and retrieving of data (Zhang, Zhu, Xu, Chen & Tran, 2018). Therefore, web mining could be used to find patterns from web content and further applied to API-based learning. Big amount of business data can be extracted with process and text mining techniques. (Ghute & Raghuwanshi, 2016) That being said, the interviews took place almost 9 months ago, which means that new priorities might have already arisen, or new ideas have been born. API ideas should be reviewed from time to time, to see if the business needs are covered and develop the API strategy if needed.

6.2 Practical implications

Practical implication	Relation to the PEC
Key stakeholder should be selected carefully based on their role, customer and API awareness. The role should inflict the desired visibility in the case company following enough knowledge from either APIs or customers.	PEC 1
API opportunities and API value propositions should be explored from time to time by auditing what kind of challenges and benefits API candidates have in order to find new API ideas regularly.	PEC 2
API ideas that arise should be documented for further use. If multiple key stakeholders or similar business needs arise, it's more likely that the API idea is beneficial. If the API idea did not gain high enough desirability in the first round it should be reanalyzed.	PEC 3
In order to define all the desirability attributes precisely a case company expert should be used to perform the analysis and further consult another expert with customer awareness	PEC 4
Desirability should be further analyzed to gain company specific weights into the priorities. API Prioritization should be made based on desirability and a company expert.	PEC 5
The decision to implement API should not be made only based on desirability. It should always be further analyzed whether the API provide the needed benefits and if any restrictions apply. If benefits are greater than penalties disadvantage the API should be implemented.	PEC 6

Table 23. Summary of practical implications

7 Conclusion

This section concludes the thesis by summarizing the research. It consists of three parts in total. The first part answers the research questions which have guided the research. The second part consists of a discussion about the limitations of the research conducted and the possible results found. Lastly, directions are given towards future research building on the discoveries that are presented.

Case company is in API discovery and experimentation stage, which is one step closer to adopting API strategy. To transit to the next phase of platform selection and targeted expansions, it is vital to identify core differentiated APIs directly central to business, only then can sticky ecosystem experiences be built and give their complete value through APIs (Holley et al., 2014). Before this leap can be made there should be roles taking responsibility about the API exploration. Even though there are now multiple API ideas constructed and prioritized, business needs can change rapidly and thus API ideas should be reviewed regularly by a group of key stakeholders and case company experts.

The next step is to define the process to maintain the APIs; when needs arise. Case company needs to build an API lifecycle management with at least API Product Manager role responsible for first making API strategy and later on the design and implementation process. Secondly, they need to start using tools and methods for understanding developers needs and preferences, like MVP, scenarios and personas (O'Neill & Golluscio, 2017). This means advancing forward towards customers in a way that customers' needs are the center of the design and development requirements. Case company also needs to create new ways for IT and business to communicate properly in lower level to deliver API Lifecycle Management to meet customers' expectations. The hierarchy of case company's IT makes it less DevOps-like fast and easily adaptable, which is why the communication channels should be developed within core resources on both sides.

A choice needs to be made by application leaders between transitioning traditional responsibilities to others while focusing all energies towards customer facing API initiatives and working with the executive team to find a leader for API business initiative with strong product management skills (Barnes et al., 2018). To gain new business activities it's important to understand the maintaining and developing costs that APIs have. If a company is prepared to invest into APIs it's probable to gain as much as 13.5 % of revenue and market value growth, but it all comes down to investing enough money and time (Still et al., 2017; Benzell et al., 2016).

7.1 Answer to research questions

API strategy is needed to help project leaders and managers to design valuable and meaningful APIs forming guidelines to follow across organization. It consists of establishing a clear vision with business objectives and building business model around API vision with detailed outlining of: costs, resources, efficiencies, value, revenue, innovation and operational process. (Ravichandran et al., 2016) After the strategy is clear the API lifecycle management should be ramped up. Specifications that should be at least defined for are: TOS, EULA, Design Guidelines, DOR, MVP and other decisions that need to be made related to architecture and design.

The designing process starts by defining API requirements and specifications in co-operation with the API consumers, identifying how API will be used and by whom as well as leveraging existing resources. (O'Neill & Golluscio, 2017; CA Technologies, 2015). The selection of the consumers, including key stakeholders should be made carefully, since the role, API awareness and customer awareness affect the quality of information gained based on PEC 1. Business need is the reason why a new system, platform, or API is needed (Alnabhan et al., 2014). PEC 2 supports existing research material suggesting that API candidates to consider are online catalogs, but the need can arise also directly from the customer. However, if the business need is not clear it can also be investigated by interviewing key stakeholders based on PEC 3. All options for implementation should be considered weather API is the right approach at all, and what is the urgency of delivery. (SOA Software, 2012) Identifying customers' needs and experiences and creating ideal customer profiles, companies need to be able to capture the key characteristics of the customer (Moilanen et al., 2018; Barnes et al., 2018).

Building face includes building client and technical environment where non-functional requirements, like security and logging should be added (SOA Software, 2012). There are multiple roles involved in the API Lifecycle Management. API Product Manager is the key role responsible for managing the team, branding, marketing, costing, tracking and billing for the API, as well as identify user profiles (O'Neill, Malinervo & Dewnarain, 2017; SOA Software, 2012). Each API should be managed independently, but within the same environment using centralized oversight and platform integration (Oliffe, 2017).

Once the strategy and API lifecycle management are built, API ideas should be gathered and prioritized based on desirability suggested by PEC 6. The more key stakeholders stand behind the API idea, the more likely it is to have higher desirability based on PEC 4. Thus, there are many techniques for prioritization, but it should always be discussed based on circumstances which one to use (Aasem, Ramzan & Jaffar, 2010). Specifically, globally significant concerns that might have an impact on requirements that affect architectural setup or the whole interface design should be carefully evaluated (Duan et al, 2009). Before moving on to production API bound to real data or backend systems, a prototype should be built because it is much less costly to make changes and it takes less time to build it (CA Technologies, 2015). When all this is done, the APIs are ready to be run, optimized and supported. (SOA Software, 2012)

Monitoring is essential in B2B API scene since it's supposed to form a central part of customer's application's functionality. Without detailed monitoring quality problems might not even occur, but instead show as decreasing usage rates. (Bermbach & Wittern, 2016) Selecting suitable key performance indicators for API is highly important, since they are always contextual factors within value chain (Leppitsch, 2018). Strategically measurements should be done in a different perspective, such as business, developer or security (Moilanen et al., 2018). APIs direct impact to company performance can best be seen in net sales only internally, whereas external impact is seen in market share and valuation. In many cases the amount of API calls or users do not correlate straight to net turnover, but more as customer loyalty and satisfaction (Benzell et al., 2017). Simple statistics like traffic rates and response times are not always good measurement, since they might inflict that a "chatty" client application has been built by developers (Barnes et al., 2018).

7.2 Limitations

This study was performed as a case study which makes some limitations to future research. It's plausible that if multiple companies participated in the study, a different outcome might have been constructed. Thus, a single case study enables to go deeper into the case and new perspectives can be found as well as singular significant findings. Also, the lack of interviewing an actual customer is a restriction, that needs to be addressed. Every interviewee has a different, individual view of what the customer needs, which is second hand knowledge and not the actual source of information.

7.3 Future research

Future research should be made in the case company as well as in the API discipline. In the case company the API ideas should be revised and reanalyzed by key stakeholders with more experience and knowledge in the case company to evaluate whether the results found in this research are reliable. In the API field, the prioritization method used, should be further tested to certify it can be used also in API field consistently. Lastly, the method to explore API ideas and analyze them should also be further tested with variations in key stakeholders' role, API and customer awareness to identify if one of the attributes is more critical.

8 Bibliography

Aasem, M., Ramzan, M. & Jaffar, A. (2010). Analysis and optimization of software requirements prioritization techniques. IEEE.

Alnabhan, M., Haboush, A., Al-Badareen, A., Al-nawayseh, M. & El-Zaghmouri, B. (2014). An Evaluation Framework for Requirements Definition of Software Development. Journal of Convergence Information Technology, 9(2): 43–50.

Barnes, H., Malinverno, P., O'Neill, M. & Santoro, J. (2018). A Strategic Marketing Mindset Is Essential to Externally Facing API Initiatives. Published 17th of April 2018. Gartner.

Baxter, P. & Jack, S. (2008). Qualitative Case Study Methodology: Study Design and Implementation for Novice Researchers. The Qualitative Report, 13(4): 544–559.

Benzell, S. Lagarda, G. & Van Alystane, M. (2017). The Impact of APIs on Firm Performance.

Berander, P. (2007). Evolving Prioritization for Software Product Management. Blekinge Institute of Technology, Doctoral Dissertation Series No. 2007:07. ISSN 1653-2090.

Bermbach, D. & Wittern, E. (2016). Benchmarking Web API Quality. DOI: 10.1007/978-3-319-38791-8_11

Billestrup, J., Stage, J., Bruun, A., Nielsen, L. & Nielsen, K. (2014). Creating and Using Personas in Software Development: Experiences from Practice. 5th International Conference on Human-Centered Software Engineering, Sep 2014, Paderborn, Germany. Springer, Lecture Notes in Computer Science Human-Centered Software Engineering, 251–258.

Bokhary, A. & Tian, J. (2018). Measuring Cloud Service APIs Quality and Usability. Int'l Conference Software Engineering Research and Practice SERP'18, pp. 208–214.

Bortenschlager, M. (2015). Building Effective API Programs: Developer Experience (DX). 3Scale. https://www.3scale.net/2015/06/building-effective-apiprograms-developer-experience-dx/ [4.12.2018]

Boyd, M. (2015). Developing the API Mindset –A guide to using Private, Partner & Public APIs, Nordic APIs. http://nordicapis.com/download.php?ebook=mindset&format=PDF

Boyd, M. (2014). Private, Partner or Public: Which API Strategy Is Best For Business? Programmable Web. http://www.programmableweb.com/news/private-partner-or-public-which-api-strategybest-business/2014/02/21 [4.12.2018]

CA Technologies. (2015). API Strategy and Architecture: A Coordinated Approach.

Cappeillo, C., Daniel, F., Matera, M. (2009). A quality model for mashup components. In: Gaedke, M., Grossniklaus, M., D´ıaz, O. (eds.) Web Engineering, pp. 236–250. Springer, Berlin.

Dayley, B. (2018). Building an Agile Application Architecture With Integrated Apps, APIs and Services. Gartner.

Dayley, B. & Oliffe, G. (2017). A Guidance Framework for Designing a Great API. Gartner.

Duan, C., Laurent, P., Cleland-Huang, J. & Kwiatkowski, C. (2009). Towards automated requirements prioritization and triage. Requirements Eng (2009) 14:73–89.

Familiar, B. (2015). Microservices, IoT, and Azure: Leveraging DevOps and Microservice Architecture to Deliver SaaS Solutions. Apress Media.

Farrag, M. & Nasr, M. (2017). A Survey of Cloud Computing Approaches, Business Opportunities, Risk Analysis and Solving Approaches. Advanced Networking and Applications 9 (2017) 2:3382–3386.

Fielding, R. & Taylor, R. (2000). Principled design of the Modern Web Architecture. Information and Computer Science. University of California, Irvine.

Fielding, R. (2000). Architectural Styles and the Design of Network-based Software Architectures. PhD Dissertation, Information and Computer Science. University of California, Irvine. 2000.

Fletcher, K. (2018). Part of Ferreira et al. Service Computing – SCC 2018. A Quality-Based Web API Selection for Mashup Development Using Affinity Propagation, 153–165.

Gamez-Diaz, A., Fernandez, P. & Ruiz-Cortes, A. (2017). An analysis of RESTful APIs offerings in the Industry.

Gao, L., Zhang, C. & Sun, L. (2011). RESTful Web of Things API in Sharing Sensor Data. IEEE.

Gebhart, M., Giessler, P. & Abeck, S. (2016). Challenges of the Digital Transformation in Software Engineering. ISCEA 2016: The Eleventh International Conference on Software Engineering Advances, 136 - 141.

Getto, G. (2014). Designing Globally Working Locally: Using Personas to Develop Online Communication Products for International Users. Communication Design Quarterly 3.1 November 2014, 24–46.

Ghute, M. & Raghuwanshi, M. (2016). Improving Business Applications Using Open Web API's. IEEE Sponsored World Conference on Futuristic Trends in Research and Innovation for Social Welfare (WCFTR'16).

Holley, K., Antoun, S., Asanjani, A., Brown, W., Costas, J., Cozzi, C. & Goyal P. et al., (2014) The Power of API Economy. IBM.

Janes, A., Remencius, T., Sillitti, A. & Succi, G. (2014). Towards Understanding of Structural Attributes of Web APIs Using Metrics Based on API Call Responses, pp. 83–92 in OSS 2014, IFIP AICT 427.

Kangas, A., Kujala, J., Heikkinen, A., Lönqvist, A., Laihonen, H. & Bethwaite, J. (2019). Leading Change in A Complex World: Transdisciplinary Perspectives.

Lenka, S., Parida, V., Sjödin, D. & Wincent, J. (2016). How digitalization capabilities enable companies to co-create value with customers. Management of Innovation and Technology, No. 2016:03.

Leppitsch, M. (2018). KPIs for APIs: 12 Key Metrics for API Programs. Apigee.com Visited 21.2.2019.

Matheny, K. (2017). A Guidance Framework for Creating Usable REST API Specifications. Gartner.

McPhee, C., Dedehayir, O. & Seppänen, M. (2017). Editorial: Platforms and Ecosystems. Technology innovation management review, 7(9): 3–5.

Meng, M., Steinhardt, S. & Schubert, A. (2018). Application Programming Interface Documentation: What Do Software Developers Want? Journal of Technical Writing and Communication 2018, 48(3):295–330.

Mohapatra, S. (2015). Requirement Management - Controlling Quality At The Upstream In Commercial Software Project Management. International Journal of Applied Engineering Research, 10(3): 8203–8219.

Moilanen, J., Niinioja, M., Seppänen, M. & Honkanen, M. (2018). API talous 101. BALTO print, Liettua 2018.

Murphy, L., Alliyu, T., Macvean, A., Kery, M. B. & Myers, B. A. (2017). Preliminary Analysis of REST API Style Guidelines. PLATEAU 17, October 23, 2017, Vancouver, CA.

Nguyen-Duc, A., Khalid, K., Bajwa, S. & Lonnestad, T. (2019). Minimum Viable Products for Internet of Things Applications: Common Pitfalls and Practices. Future Internet 2019, 11:50.

Olliffe, G. (2018). Decision Point for API and Service Implementation Architecture. Gartner.

Oliffe, G. (2017). A Guidance Framework for Evaluating API Management Solutions. Gartner.

O'Neil, M. & Golluscio, E. (2017). Master B2B Ecosystem Integration With a Blended EDI and API Approach. Gartner.

O'Neil, M., Malinverno, P. & Dewnarain, G. (2017). Choosing the right API Pricing (and Funding) Model. Gartner.

Otero, C., Dell, E., Qureshi, A. & Otero, L. (2010). Quality-Based Requirement Prioritization Framework Using Binary Inputs. 2010 Fourth Asia International Conference of Mathematical/Analytical Modelling and Computer Simulation.

Papazoglou, M. & van den Heuvel, W. J. (2006). Service-Oriented Design and Development Methodology. Int. J. of Web Engineering and Technology (IJWET), 2006.

Rama, G. & Kak, A. (2013). Some Structural Measures of API Usability. Published online in Wiley Online Library: wileyonlinelibrary.com. DOI: 10.1002/spe.2215.

Ravichandran, A., Taylor, K. & Waterhouse. P. (2016). DevOps for Digital Leaders: Reignite Business with a Modern DevOps-Enabled Software Factory. CA, Inc.

Ries, E. (2011). The Lean Startup. How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Business.

Ries, E. (2014). The Lean Startup: How Constant Innovation Creates Radically Successful Businesses; Penguin Group: London, UK, 2014.

Robillard, M. (2009) What Makes APIs Hard to Learn? Answers from Developers. IEEE Software November/December 2009, 27–34.

Schneider, S., Wollersheim, J., Krcmar, H. & Sunyaev, A. (2018). How do Requirements Evolve Over Time? A Case Study Investigating the Role of Context and Experiences in the Evolution of Enterprise Software Requirements. Journal of Information Technology (2018), 33: 151–170.

Smith, G., Ofe, H. A. & Sandberg, J. (2016). Digital service innovation from open data: Exploring the value proposition of of an open data marketplace. In system science (HICSS), 2016 49th Hawaii International conference on (1277 –1286). → 41.

SOA Software, Inc. (2012). Enterprise API Platform - Building Successful APIs. url: http://docs.akana.com/sp/assets/Building_Successful_APIs.pdf

Still, K., Seppänen, M., Seppälä, T., Suominen, A., Valkokari, K. & Korhonen, H. (2017). ”Alustatalous on vuorovaikutustaloutta”. ETLA Muistio No 61, 15.9.2017. http://pub.etla.fi/ETLA-Muistio-Brief-61.pdf

Stylos, J. & Myers, B. (2007). Mapping the Space of API Design Decisions. IEEE Symposium on Visual Languages and Human-Centric Computing.

Vijayakumar, T. (2018). Practical API Architecture and Development with Azure and AWS: Design and Implementation of APIs for the cloud.

Wulf, J. & Blohm, I. (2017). Service Innovation through Application Programming Interfaces - Towards a Typology of Service Design. Thirty Eighth International Conference on Information Systems, Seoul 2017.

Zhang, Y., Zhu, L., Xu, X., Chen, S. & Tran, A. B. (2018). Service Computing - SCC 2018. Data Service API Design for Data Analytics, 87–102.

Yazan, B. (2015). Three Approaches to Case Study Methods in Education: Yin, Merriam, and Stake. The Qualitative Report, 20(2): 134–152.

Yin, R. (1993). Case Study Research - Design and Methods, second edition. Applied Social Research Methods Series Volume 5. Sage Publications.