Jyväskylän yliopiston julkaisuarkisto
Jyväskylä University Digital Archive

UNIVERSITY OF JYVÄSKYLÄ

This is an electronic reprint of the original article.
This reprint *may differ* from the original in pagination and typographic detail.

Author(s): Järvinen, Janne; Huomo, Tua; Mikkonen, Tommi; Tyrväinen, Pasi

Title: From Agile Software Development to Mercury Business

Year: 2014

Version:

Please cite the original version:

Järvinen, J., Huomo, T., Mikkonen, T., & Tyrväinen, P. (2014). From Agile Software Development to Mercury Business. In C. Lassenius, & K. Smolander (Eds.), Software Business. Towards Continuous Value Delivery : 5th International Conference, ICSOB 2014, Paphos, Cyprus, June 16-18, 2014. Proceedings (pp. 58-71). Springer. Lecture Notes in Business Information Processing, 182. https://doi.org/10.1007/978-3-319-08738-2_5

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

From Agile Software Development to Mercury Business

Janne Järvinen[1], Tua Huomo[2], Tommi Mikkonen[3], and Pasi Tyrväinen[4]

[1]F-Secure, Helsinki, Finland
janne.jarvinen@f-secure.com

[2]VTT, Oulu, Finland
tua.huomo@vtt.fi

[3]Tampere University of Technology, Tampere, Finland
tommi.mikkonen@tut.fi

[4]University of Jyväskylä, Jyväskylä, Finland
pasi.tyrvainen@jyu.fi

Abstract. The rapid downfall of the Nokia software ecosystem has radically altered the landscape of software industry in Finland in recent years. There has been a shift from largely corporate driven way of working, which is often dominant in large companies, to more agile practices, and in general software organizations are seeking new ways of composing, delivering, and using software also inside already established companies. To accelerate this transformation in large scale, a collaborative research program has been created, called Need for Speed (N4S). In this paper, we give an insight to the joint goals and concrete actions of the program and discuss the motivations of individual companies that are participating in the program. As one concrete goal of the project, we introduce the concept of Mercury business, where the principles of the Lean startup framework are applied in a more conventional industrial setting.

Keywords: Real-time value delivery, deep customer insight, lean startup, elastic enterprise, mercury business.

1 Introduction

The rapid downfall of the Nokia software ecosystem has radically altered the landscape of software industry in Finland in recent years. Instead of a single ecosystem that has been aiming at the creation of software – including hardware elements, low-level software, operating systems, middleware, and applications – for mobile phones, where major up-front R&D investment has been the norm, smaller companies as well as startups are now becoming major actors. This change has meant that new ways of composing, delivering, and using software are emerging, following the spirit of e.g. the Lean Startup framework [11].

While newly founded companies may find it easy to start operating in accordance to the Lean Startup ideals, already established companies, especially those that have been formerly a part of the Nokia ecosystem, have been operating with a different

mindset. They have been largely focusing on creating software products, which have been delivered in a somewhat traditional fashion, and this is also reflected in their processes and organization, which complicates entering new markets and experimenting with new products.

To create the foundation for the future success of the Finnish software intensive businesses in the new digital economy, largely fueled by the Web and pervasive connectivity in almost all places, a collaborative, industry driven research program has been created, called Need for Speed (N4S) [8]. The project is planned for years 2014-2017, and its budget exceeds 80M€, resulting in an annual budget around 20M€. The program is executed jointly by the industry and academia, and it presently is the biggest national investment in software-related research.

In this paper, we give an insight to the joint goals and concrete actions of the program and discuss the motivations of individual companies that are participating in the program. Moreover, we will also introduce research actions that will be executed during the first year of the program. As an additional contribution of the paper, we address a concrete business goal of the project, so-called Mercury business, where the principles of the Lean startup framework are applied in a more conventional industrial setting. Topics such as internal startups and elastic enterprises are also closely related to our approach.

The rest of this paper is structured as follows. In Section 2, we address agile and lean software development, which reflects the general state-of-the-practice in Finnish (and to a great extent also Global [10]) companies, although there are small deviations [5]. In addition, we provide some background information regarding new software and business approaches that are applicable in the N4S setting. In Section 3, we introduce the goals of N4S, and discuss each research goal separately. Moreover, we also show the big picture of these goals to demonstrate the changes we are aiming at. In Section 4, we give an extended discussion on the business impacts of the program, together with a case study that was already executed to study the feasibility of the approach. In Section 5 we give an extended discussion on our observations so far in the creation of the consortium for the program, as well as point out certain important details. In Section 6, we draw some final conclusions.

2 Background

In the following, we first discuss contemporary software development approaches that are commonly applied in Finnish software companies. Then, we address disruptive technologies that challenge the old ways of working in the field of software. Finally, we briefly introduce the Lean startup approach, which has been an inspiration during the planning of the N4S program.

2.1 Agile and Lean Software Development

Software and software intensive industry have undergone major advances over the last decades. The transition from slow projects lasting years to the rapid cycles of continuous development and deployment have been dramatic (Fig. 1).

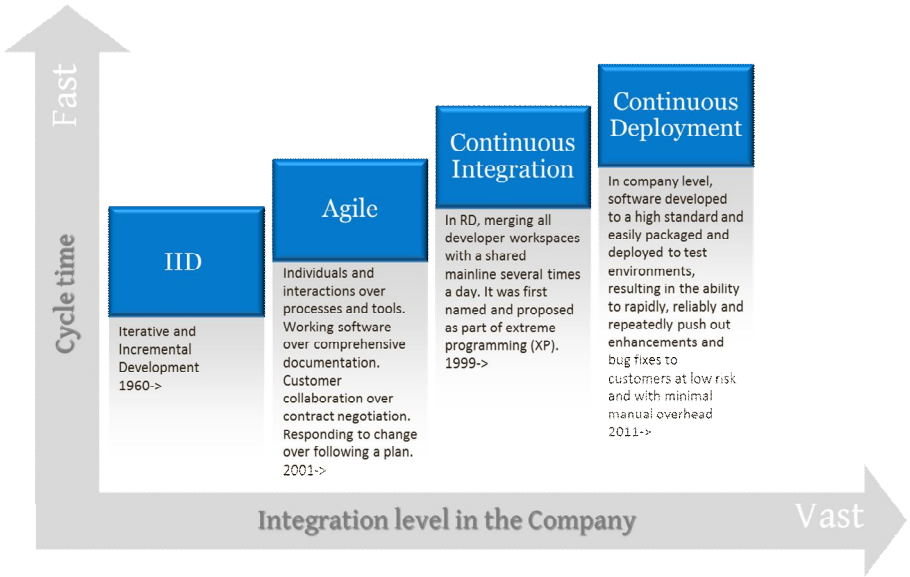

Fig. 1. Agile and lean software development.

Iterative and incremental development. Ever since (and probably even before) the introduction of commonly misunderstood Waterfall process [12], iterative and incremental development has been used by software developers to manage risks and uncertainties in software development. By developing software in a piecemeal fashion, where frequent checkpoints can be used to detect anomalies and misinterpretations, the development effort can be more easily managed than by using a big-bang development approach. Consequently, while the rational design process can be used to explain how the development advances [9], in reality it has been customary to conduct at least experiments before advancing too far in the development.

Agile development. In many ways culminating in the Agile Manifesto (http://agilemanifesto.org/), agile software development approaches [3] consist of a wide number of practices where delivering value to a customer is the dominant factor in software development, over following a plan, which had been the prevailing concept early on in many software projects. Various agile methodologies exist, including Extreme Programming [2], Scrum [13], Kanban [1], and Lean software development, which more or less share the underlying mindset but implement the actual actions differently.

Continuous integration. When numerous developers work on the same project, they commonly make changes in the same software components in their own workspaces. When the changes contradict each other, a conflict arises, which need to be resolved by the developers. The key issue of continuous integration is to minimize such conflicts by merging developer workspaces with a shared mainline [6]. Whenever a change to the mainline is made, the whole system is compiled and an automated test run is made to ensure that the mainline remains healthy. It is important

to notice that continuous integration is a development related issue, and therefore it mainly concerns R&D of software organizations.

Continuous deployment. While continuous integration is about creating the ability to build a system automatically when even a smallest change has been made, continuous deployment is about creating the ability to deliver the smallest added value to the customers. Obviously, to minimize risks this requires automating all the processes that must be executed to deliver the software to customers, and therefore implementing continuous deployment concerns the whole company.

To summarize, the evolution of the software development approaches has led towards approaches where the step between the development and deployment is being reduced. Hence, an approach referred to *DevOps* emerges, where development is treated similarly to operations, and no distinction between the two is made. Thus, DevOps stresses communication, collaboration and integration between software developers and information technology (IT) professionals responsible for the operation of the information systems [4]. The promise is that the tighter cooperation results in rapid development and utilization of the software products and services. To reach this target it is common that continuous deployment and/or continuous delivery [7] are used. Moreover, in order to gain benefits from the capability to release rapidly requires that also business goals are defined in a clear and achievable fashion.

2.2 The New Operating Environment

The Internet has rapidly become far more pervasive than it was only a few years ago. At present its transformational effects are spreading into several sectors of the economy and society via new innovations, services, and the emergence and quick success of new companies.

The complexity and competition around the new Internet infrastructure, services and business environment will increase dramatically which will fundamentally change the way software will be developed, deployed and used to reach business goals. The Internet partly already is and will increasingly be the first truly global platform for the digital economy. It will enable significant new business, economic and social opportunities. Consequently, we are facing a fundamental systemic transformations towards a world where digital resources are constantly available on-line, and available for all to use.

These systemic transformations will take many forms. Increasingly, products and services are not developed by a single company but rather by a network of collaborating companies. New, still partially emerging ecosystems and new competitors will alter industry structures, the public sector, supply chains and many other aspects of today's businesses. Similarly, computing and networking infrastructures, approaches, and processes have changed dramatically over the last years – faster and faster networks, cloud and web technologies, open source, Internet of Things, and open data approaches – and they are all reshaping the digital economy in unforeseen ways and scale.

These new opportunities are increasing ability to gather feedback regarding the use of products, customer satisfaction, and various other aspects that we have commonly overlooked. Such methods are already commonly used in today's software systems to

e.g. report bugs – something that is fundamentally associated with software development. However, in the future, there will be similar facilities for other use to help understanding how customers are using products and to create models regarding why. The central concept in the new internet economy is the idea of a minimum viable product or service, which aims at defining the smallest possible implementation that brings added value to customers. Upon delivering the product or service, the focus shifts to creating incremental improvements, so that development cycles can be shortened, progress can be evaluated, and customer feedback and insight can be used to measure the value of the improvement and fed back to development in real-time. Today, game and web service companies are already leading the way towards deep customer understanding to improve gaming experience or to help in e.g. selecting suitable advertisements to show, but we expect that many other fields of computing will be quick to follow. When combined with the ability to rapidly scale operations, the concept resembles that of *elastic enterprises* [14].

2.3 The Lean Startup

In the Lean Startup framework [11], so-called Minimum Viable Product (MVP) is a key concept. With MVP the developing organization can find the critical and most valuable features with the customers by experimenting with new iterations in the market.

The core of the Lean Startup approach is to execute a build-measure-learn cycle iteratively. These activities are linked to artifacts, ideas, product, and data. In each iteration, ideas are transformed to products by building them, then, as a product is used, usage patterns are measured, and finally measurement data is used to learn new ideas. The goal of these iterations is to learn what features customers are ready to pay for, and which are not interesting for them.

Customer development is an essential activity in the Lean Startup approach, and it defines how the Lean Startup approach is applied as the company starts to grow. There are four phases that follow each other, 1) customer discovery, 2) customer validation, 3) customer creation, and 4) company building. The goal of customer discovery is to test both problem and product hypothesis, which the customers would like to be solved. Once the hypotheses have been approved, in customer validation phase, the goal is to create a sales roadmap, with a sales cycle that is feasible. It is possible to iterate between these two phases; agile tactics, such as short releases, simple designs and refactoring commonly play a role in these steps. When the product is good enough, the remaining two phases are executed in more traditional fashion, where business plans based on the product are created and usual market creation activities are executed.

While the lean startup approach defines no particular process or tools that are to be used in software development, in general agile development approaches are assumed to minimize the time from a concept to a prototype that can be experimented with. As for the analysis, basic statistical methods and measures related to business goals are used to study whether the desired results are achieved with the existing implementation. Moreover, so called A/B testing, where different versions are tested

in parallel, and the version that is best received by customers will be selected for future use and development, is often applied in this context.

3 Towards Mercury Business

The N4S program has been built around three main themes. These are 1) paradigm change from product business to delivering value at real-time; 2) deep customer insight to improve the hit-rate of businesses; and 3) Mercury business which explicitly aims at finding the new money instead of focusing only on the traditional customers. All these goals build on established practices presented in Fig. 1 earlier, but this time the focus has been shifted from software development view to the business impact created with software.

In the following, these three goals, which are also illustrated in Fig. 2, will be addressed separately. However it is important to notice that all of them jointly enable the new breed of software business we refer to Mercury business as described in [8].

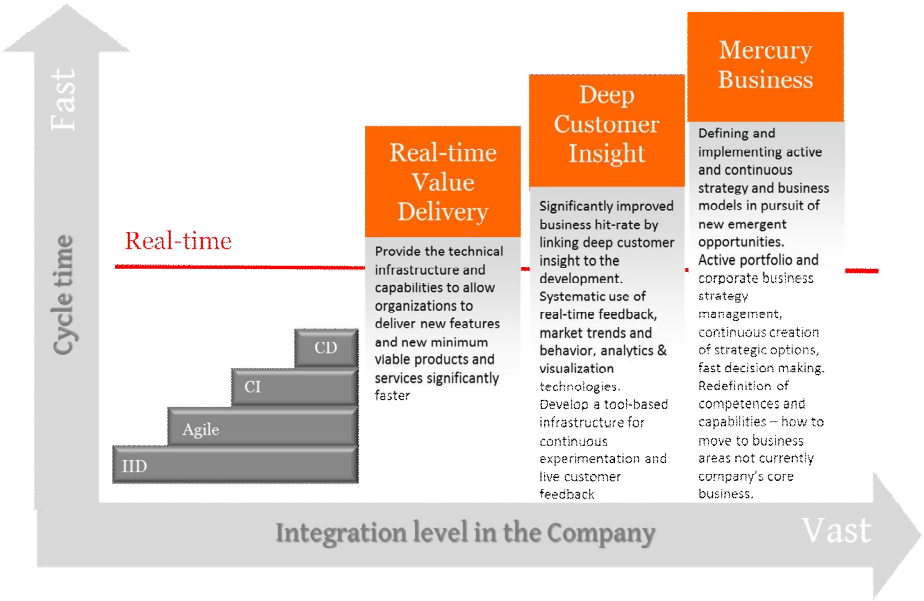

Fig. 2. Real-time value delivery, deep customer insight, and mercury business.

3.1 Real-Time Value Delivery

The key aspect of the N4S program is to catalyze a paradigm change from the traditional product-based software business to service-based business where value can be delivered at near real time. Achieving this goal requires careful reconsideration of the mode of operation as well as seamless integration of businesses and research and

development – the former provides motivation for the latter, whereas the latter enables new forms of business. Obviously, also technical infrastructure and required capabilities must be established to support the transformation. In addition, special attention is required to maintain the present level of quality, or, better yet, improve the quality experienced by customers by focusing on fewer features but delivering them more rapidly.

To reach the above goals, an architecture that supports the incremental development of systems is needed, where features can be added and removed easily. Moreover, this architecture must be complemented by a continuous integration system that can build and test new versions, implying that automatic and incremental generation of test cases and interpretation of test results are a necessity. Finally, deployment of the software must also be automated, with mechanisms to minimize or eliminate downtime and inconveniences for the users.

3.2 Deep Customer Insight

The goal of deep customer insight is to invent value-creating solutions, and act as a source of inspiration for new products, features, or services that create customer value, which typically stems from the customer contexts and not from the engineering domain. The goal is to quickly gain and assess information regarding the true customer value of potential services, product features, and other possible aspects of user interaction with a service or a product. As a prerequisite, understanding of customer contexts and development opportunities as well as an insight on the ways how customers live and work are needed. The deep understanding of the customers, usage of products and rapid feedback are gathered continuously from the live use of the products, and any possible weak signals.

Conducting live experiments enable studying how the users actually interact with a service or a product. However, successful collection of usage data requires understanding regarding what data to collect. Data that is readily available and simple to collect does not necessarily lend itself to meaningful interpretation in terms of what can be related to the user value of the features or true needs of the user. Therefore, before running the experiments, these experiments should have a defined scope and purpose. One can start the experimentation from simple features and interactions, but the ultimate goal of the program is also to enable experimenting and testing ideas and concepts early in the development – not only after the product or service or hardware for the product exists.

To achieve the above goals, there is a thriving demand for automatic and efficient feedback systems, analytics and visualization. The potential of efficient feedback systems and analytics of different flavors is huge, as companies realize the importance of understanding their customers' cultural differences and behavior in different situations.

3.3 Mercury Business

By Mercury business, we refer to companies and societies being able to behave like "mercury" finding new grooves where to flow to grow new business. The goal is to enable companies to actively seek new ways to execute their existing businesses, and, perhaps even more importantly, also experiment the options to transform themselves to completely new business areas. The two above goals, real-time value delivery and deep customer insight, are important prerequisites for Mercury business, but there are also other factors that must be considered. For instance, it is obvious that company culture, structure, and leadership must be altered – from individuals and going all the way to organizational structures – to empower everyone to seek new opportunities. Indeed, one important factor is extreme organizational flexibility, where all kinds of changes are made culturally as easy as possible. The ways of working may also change dynamically regardless of the existing organizational structures. These changes are possible e.g. in the Finnish individualistic culture, where extremely dynamical changes in the ways of working are possible.

Finally, while the Mercury business model may change existing products and portfolios, we believe that its ability to totally convert the company into a new business domain is more important. This is what we believe will be an important characteristic for the next-generation software business even in the global scale.

Lean Startup vs. Mercury Business. As already mentioned, Mercury business is closely related to the Lean startup framework, and in many ways the Lean startup has been an inspiration for Mercury business. However, while Lean startup is about the creation of a new company and the definition of its products, Mercury business aims at transforming and extending already existing businesses, which requires a different approach. The main differences between the Lean startup and Mercury business are listed in Table 1.

Table 1. Lean startup vs. Mercury business.

Lean startup	Mercury business
No rigid organization; emerging company that is seeking for a form.	Already existing organization that seeks new markets and opportunities; internal startups can be used to separate new effort from already existing business.
Experiment potential products that could be scalable to different markets.	Experiment scaling of existing products (or product derivatives) to new markets, experiment scaling of features in existing products.
Rapid pivoting where old products can be abandoned for better ones.	Whole experiment is about experimenting new opportunities; existing products and markets not risked.
Usually only one product at a time is being considered.	Numerous parallel experiments are possible.
No existing infrastructure for supporting experimenting; built as a part of the product and the experiment.	Established infrastructure for experimenting must be in place.
Build-measure-learn.	Measure-learn-build.

5 Thematic Analysis of N4S Cases

From program management perspective the work in the N4S program has been divided to 1-4 cases per participating firm. Each of the cases has a case owner from the firm, a research coordinator from a research institution and one or several firms and research institutions working on the tasks related to the case. The cases are expected to impact the participating firm performance in line with the targets of the program.

For the purposes of analyzing the 49 cases defined in the beginning of the program for the 26 firms, the program preparation team analyzed the case descriptions provided by the participating firms, extracted key concepts from them and annotated them with on an average three labels. For focused cases one label described well the connection of the case to the targets of the program while some broad cases needed up to six theme labels. Total 23 labels were used the most common ones being as follows. In Mercury Business area: Mercury business model trials (10), Partnering approaches (6), Skills and capabilities (6), and New market opportunity / domain detection (6). In Deep Customer Insight area: Fast feedback / voice of customer (15), Telemetrics – data analysis (10), Customer and business landscape analysis (8), Multidimensional segmentation (6), and Experimentation culture (6). In Real-Time Value Delivery / Continuous Deployment area: Real-time value delivery tooling (21 cases), real-time value delivery (21), and Variability and reuse management (10).

The target of thematic analysis is to identify, which business cases can be clustered together based on shared themes. For this purpose the annotations the 49 business cases were compared by two means. First, the business cases sharing two or more themes were connected together and the resulting graph created with GVisualize is presented in Figure 3. Secondly, the thematic labeling of each case was treated as a vector in a 23 dimensional space. Figure 4 represents connections between cases, whose vector multiplication exceed a threshold value of 0,30.

From the first graph we can identify four clusters of cases:

- Real-time value delivery and real-time value delivery tooling. This is the largest cluster and includes the cases, where the main emphasis is enabling and automating the continuous deployment and value delivery. This represents the first step in building the capabilities of the firm in the program themes.
- Fast customer feedback and data analysis. The cases in this cluster are making use of the data collected from the customer for the second step of the program, i.e. for gaining deep customer insight.
- Customer and business landscape analysis. These cases operate in between the deep customer analysis and mercury business targets of the program. They connect the landscape analysis either with the fast feedback or with mercury business model trials.
- Mercury business models and variability / reuse management. These cases combine creation of mercury business models to the context of product variability and target in creating new business from this combination.

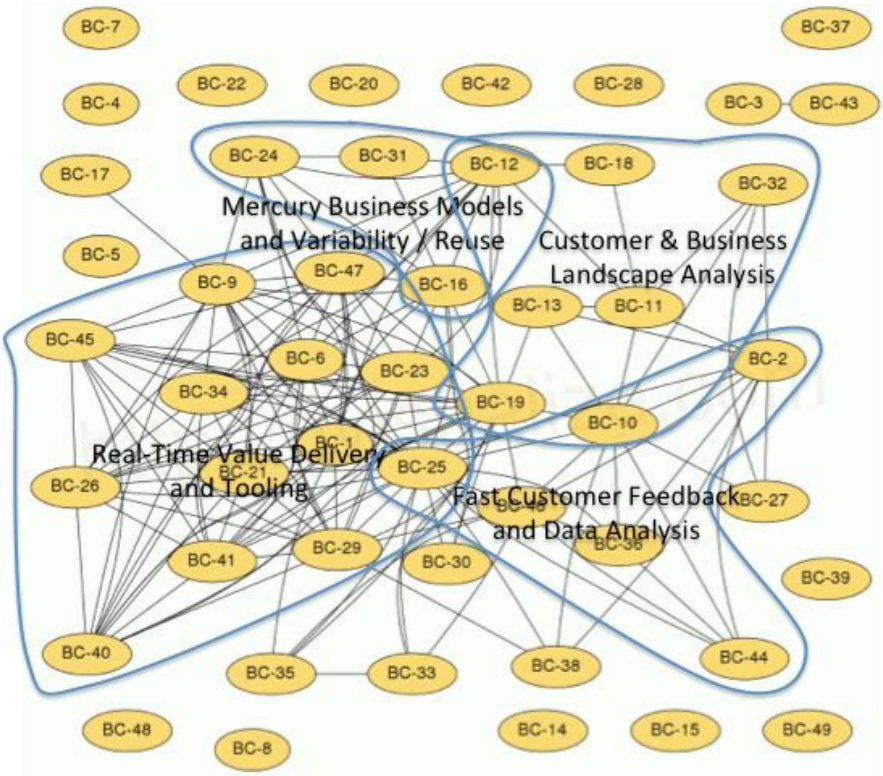

Fig. 3. Cases for year 2014 clustered based on sharing two or more common themes.

In addition to the four clusters in Figure 3, there are some cases connected with either the real-time value delivery or tooling and customer feedback (see cases 30, 33, 35 and 38) while they do not form a thematically uniform cluster to the extent the four clusters presented here.

Figure 4 shows three major clusters, which are formed around three key labels, Real-time value delivery tooling, Fast feedback / Voice of Customer and Mercury business trials. These themes were among the most used thematic labels and also parts of the connected themes in Fig 3.

Due to use of the vector multiplication and a high threshold value (0,30) the cases having only one theme label tend to form the core of the clusters. Instead, cases with several thematic labels tend not to match with several labels of another case. And vice versa, cases 15, 22 and 39 in the middle of the clusters in Fig 4 are not connected in Fig 3 as they have only one thematic label.

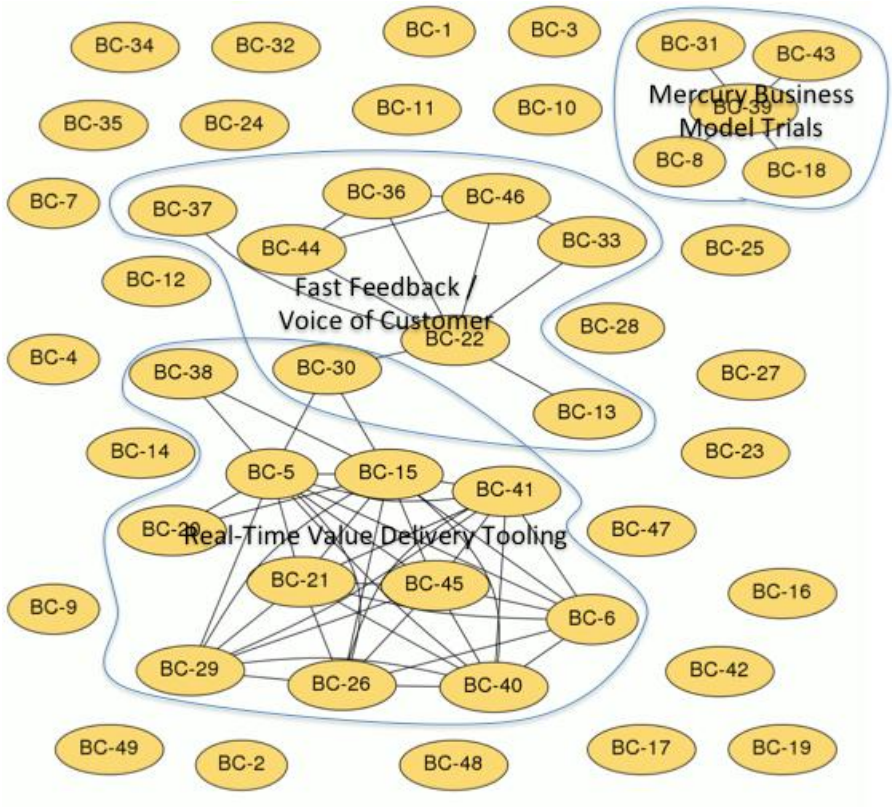

Fig. 4. Cases for year 2014 clustered based on vector multiplication value exceeding 0,30.

From the perspective of this paper the empirical part brought up the key themes within the three main themes. Quantitatively we also noticed that majority of the cases start with the first main theme with real-time value delivery, followed by cases focusing on using fast customer feedback and data analysis as the means to gain deep customer insight. In area of mercury business the mercury business model trials were connected with customer/business landscape analysis or variability/reuse management. In addition, the main theme on Mercury business included several cases focusing on partnering approaches, skills and capabilities, and new market opportunity / domain detection.

5 Discussion

The increasing interest in seeking new markets to face increasing competition requires lean approach to numerous operations. Contemporary software development ideologies – such as Scrum, Kanban, Lean production, and DevOps mentioned above, are building on the possibility to perform small changes that are delivered to the customers as soon as they are completed. While this delivery does not need to take

place immediately as the new features are completed, the option to do so is of pivotal importance in Mercury business, as the decision regarding the deployment can be made based on markets rather than technical competences and capabilities – in other words the technical capabilities are extended to business operations. Similarly to software development, the ability to execute new business does not necessarily mean that actions should be taken immediately, but for obvious reasons, such as advertising campaigns, the exact time to go live may be a subject to a strategic, company level decision.

The execution of Mercury business builds on some of the characteristics of the elastic enterprise [14], where five key dynamic properties have been identified that help in scaling businesses in aggressive fashion. These are business platforms, business ecosystems, universal connectors, cloud infrastructure, and sapient leadership. In Mercury business, each business attempt still builds on these properties, but the attempts should be framed as a live experiment. In particular, there must explicit goals that determine whether or not it makes sense to continue the attempt to create new business through scaling the existing business and technical infrastructure.

To summarize, the most important differences between Mercury business and Lean startup arise from the fact that in an already established company, there commonly are assets that the company seeks to benefit from also in the future. Identifying the way and the domain in which the assets become valuable are the key issue of Mercury business. By contrast, in the Lean startup approach, the key question is what assets to build. Another difference is that while a company with existing business can extend its resources to various parallel experiments, in the creation of a startup the focus is commonly placed on the most important aspect. Our claim is that the cost of these experiments will be significantly reduced with the help of highly automated infrastructure providing capability of real-time value delivery with deep customer insight. What is common in both approaches are the elements regarding scalability of assets, be it those that a new company will build or those that already exist. We believe that scaling is the fundamental key characteristic of all successful Internet era businesses.

6 Conclusions

The newly emerged Internet based business operating environment is paving the way towards a new world of business. These days, everyday artifacts and services such as documents, photos, music, videos and newspapers are widely available on the Web. Online banking and stock trading have become commonplace. Various documents that used to be difficult to access, such as municipal zoning documents, government budget documents or tax records, are now readily available on the Web.

To deal with this change, many companies are at the brink of a major shift on how they define their next-generation competitive strategy, new leadership approach and operating processes that would form a strong basis for changing economic conditions. The key question is how the companies could adapt to radically new business conditions and opportunities in real-time or even proactively.

The quantum leap in software development speed by incrementally building and deploying software with real-time customer feedback will facilitate the speed and flexibility needed in the Internet-time business competencies. Perhaps paradoxically, software development, which has sometimes been criticized for slowing down the business, has become a source for rapid innovations. Harnessing this ability to serve strategic business intents requires drastically new approaches. The transformation and radical rethinking which takes companies into totally new markets and enables them to benefit from the most viable business opportunities, are built on the concepts such as new strategic thinking and leadership, rapid development cycles, validated learning, scientific, but cheap live experimentation, and iterative releases with minimum viable products and services.

References

1. David Anderson. Kanban – Successful Evolutionary Change for your Technology Business. Blue Hole Press, April 2010
2. Kent Beck. Extreme Programming Explained, Second Edition, Addison-Wesley Professional, 1999
3. Cockburn, Alistair. Agile Software Development, 1st edition, December 2001, 256 pages. Addison-Wesley Professional.
4. Debois, P. Devops: A software revolution in the making, Cutter IT Journal. Vol 24, No 8, August 2011
5. Eloranta, V.-P., Mikkonen, T. Koskimies, K., and Vuorinen J. Scrum Anti-Patterns: An Empirical Study. Accepted to APSEC'13, to appear.
6. Fowler, M. Continuous Integration, 2006. http://martinfowler.com/articles/continuousIntegration.html
7. Humble, J. and Farley, D. Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education, Jul 27, 2010.
8. Huomo, T., Järvinen, J., Kettunen, P., Kuvaja, P., Koivisto, A., Lassenius, C., Lehtovuori, P., Lilja, S., Miettinen, S., Mikkonen, T., Münch, J., Männistö, T., Oivo, M., Partanen, J., Porres, I., Still, J., and Tyrväinen, P. Strategic Research Agenda for Need for Speed. ICT SHOK DIGILE, April 22, 2013. Available at http://www.digile.fi/Services/researchprograms/futureprograms. Last referenced Jan 2014.
9. D L Parnas and P C Clements. A rational design process: How and why to fake it. IEEE Trans. Softw. Eng. 12, 2 (February 1986), 251-257, 1986.
10. Pikkarainen, M, Haikara, J., Salo, O. Abrahamsson, P. Still, J., The impact of agile practices on communication in software development. Journal of Empirical Software Engineering, Vol. 13, number 3, pp. 303-337 (2008).
11. Ries, E. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Publishing Group, 2011.
12. Royce, Winston (1970), Managing the Development of Large Software Systems, Proceedings of IEEE WESCON 26 (August): 1–9
13. Ken Schwaber. Scrum Development Process, In Business object design and implementation: OOPSLA '95 Workshop Proceedings. The University of Michigan. p. 118.
14. Vitalari, N. and Shaughnessy, H. The Elastic Enterprise: The New Manifesto for Business Revolution. Telemachus Press, 2012.