AGILE METHODOLOGIES IN PRODUCT DEVELOPMENT

CASE: DISCRETE MANUFACTURING AND ASSEMBLY

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Master's thesis

Visamäki, Strategic Leading of Technology-based Business

Spring 2019

Tero Lappalainen

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Strategic Leading of Technology-based Business
Visamäki

Author	Tero Lappalainen	**Year** 2019
Title	Agile Methodologies in Product Development Case: Discrete Manufacturing and Assembly	
Supervisor(s)	Jukka Pulkkinen	

ABSTRACT

During last years, the most successful ICT companies has adopted Agile methodologies. What happens when these methods are used in totally different industry? Rapid product development cycles and managing continuously changing priorities are key attributes in developing highly customized products. Aim of this action research was to understand, whether Discrete Manufacturing and Assembly Company's product development process required changes, could design work benefit from agile practices, how new operation model can be implemented and what kind of results it will deliver.

Qualitative semi-structured interviews were used to understand current situation and requirement for change. In parallel, a literature and media study were made of methodologies with highest adoption and growth, and industry specific applications. Based on this information, a framework comparison according subject’s prerequisites and a recommendation of most suitable frameworks was made. This was followed by a new operating model description, implementation of required tools and introduction of new model.

After and before surveys and comparison of design release volumes for past two years was conducted to evaluate the change impact. Based on the results, a following conclusion was made. Using agile methodology and practices led into increased process clarity, better support for design of high-quality products and increased transparency and communication perceived by process contributors. There was no sound evidence for productivity gains, as effects of other factors could not be ruled out. However, gains through continuous improvement cannot be excluded.

Keywords Agile methodologies, Lean management, product development

Pages 79 pages including appendices 15 pages

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Teknologiaosaamisen johtaminen
Visamäki

Tekijä	Tero Lappalainen	**Vuosi** 2019
Työn nimi	Agile Methodologies in Product Development Case: Discrete Manufacturing and Assembly	
Työn ohjaaja(t)	Jukka Pulkkinen	

TIIVISTELMÄ

Poikkeuksetta viime vuosien menestyneimmät ICT-yritykset ovat hyödyntäneet tuotekehityksessään ketteriä menetelmiä. Miten käy, kun samoja menetelmiä käytetään kokonaan toisella toimialalla? Lyhyiden, kustomoitujen erien tuotannossa, on tuotekehityssyklien nopeus ja jatkuvasti muuttuvien prioriteettien hallinta äärimmäisen tärkeää. Kehittämistutkimuksen tavoitteena oli selvittää, onko Discrete Manufacturing and Assemblyn tuotekehitysprosessia tarve muuttaa, miten ketteriä menetelmiä voidaan hyödyntää suunnittelutyössä, miten uusi toimintamalli otetaan käyttöön ja mitä tuloksia sillä voidaan saavuttaa.

Nykytilanteen ja muutostarpeen selvittämiseksi hyödynnettiin laadullista haastattelututkimusta. Samanaikaisesti perehdyttiin käytetyimpiin ja viime aikoina eniten käyttäjäkuntaansa kasvattaneisiin ketteriin menetelmiin ja toimialan sovellutuksiin hyödyntäen lähdekirjallisuutta ja verkkomateriaaleja. Näiden pohjalta luotiin viitekehysten vertailukehikko, jossa huomioitiin toimeksiantajan tarpeet ja tehtiin suositus sopivimmista viitekehyksistä. Tämän jälkeen tehtiin uuden toimintavan kuvaus, toteutettiin tarvittavat työkalut ja otettiin uusi tapa käyttöön.

Vaikutusta mitattiin muutosta ennen ja jälkeen tehdyillä määrällisillä verkkokyselyillä ja vertaamalla suunnitelmien julkaisumääriä kahden edellisen vuoden tuloksiin. Lopputuloksena ketterien menetelmien hyödyntäminen kasvatti prosessin osallistujien mielestä prosessin selkeyttä, tuki paremmin laadukkaiden tuotteiden suunnittelua, teki priorisoinnista ja kommunikaatiosta läpinäkyvämpää. Tuottavuuden kasvulle ei tutkimuksen aikana löydetty aukottomia todisteita, vaan muiden tekijöiden vaikutus jäi merkittävämmäksi. Tämä ei kuitenkaan poissulje uuden mallin mukaisen jatkuvan kehittämisen kautta syntyvää kasvua jatkossa.

Avainsanat Ketterät menetelmät, Lean-johtaminen, tuotekehitys

Sivut 79 sivua, joista liitteitä 15 sivua

CONTENTS

APPENDICES

1 INTRODUCTION

During the summer 2018 I had a discussion with a member of Discrete Manufacturing and Assembly Company about Agile methodologies and how well those transfer outside IT domain. Having 15 years of experience of leading technology teams and ten years' experience of Lean adoption and several years about Agile methodologies, I still felt underqualified to make any strong recommendations or prescriptions. As the person was aware that I was finishing my studies, a question was raised whether I would be interested to make a case study of such implementation. Being horrified about the idea, as I didn't know practically anything about developing physical products, but at the same time intrigued about the possibility to test these methodologies outside my comfort zone in such a greenfield environment induced me to take the assignment.

Discrete Manufacturing and Assembly Company (DMA) has grown significantly during past couple of years from virtually zero to a 130-person organisation comprising almost hundred million euros revenue. Results have been promising, but at the same time need for more structured approach to ensure transparency of priorities, constant communication and cumulative learning of the team has surged. On the other hand, traditional structures and management models used in manufacturing industry are perceived too bureaucratic and rigid, eventually killing agility, which is seen as one of the critical success factors. Could emergent Lean-based Agile methodologies be the solution?

Agile methodologies and practices have been used in IT industry for over quarter of century and today it is claimed that they are used by 80–97% of organisations (CA 2018; VersionOne 2018) to some extent. Unfortunately, most of the surveys are made or sponsored by IT industry companies and independent research is marginally available. Many of the prosperous newcomers in technology industry like Amazon, Facebook, Google, Netflix and Spotify have openly told that they use Agile methodologies and has been actively participating in Agile conferences and publishing material of the Agile implementations.

Of course, business agility and being able to rapidly answer to increasing competition and changing customer needs sounds compelling and who would not like to be compared to these technology giants. Again, academic, peer reviewed quantitative surveys, not to mention longitudinal research, is sparse and most of the academic research has been single-case studies. As an exception Serrador and Pinto (2015, 1047–1049) made a research about Agile benefits with a conclusion that Agile methodologies increase efficiency, stakeholder satisfaction and perception of overall project performance. One of the largest industry surveys of 3000 projects has been conducted by Reifer (2017) suggesting 7–12 % productivity gains, 5–12 % lower unit costs and 6–12 % increased quality. There is also 75–90% project schedule achievement compared to 40–60% resulted from following traditional project management methodologies, though this has been achieved by delivering fewer features.

Tempting figures but adopting Agile Software development methodologies and practices is still rare outside ICT industry. Although, Scrum for Hardware (Brown & Justice, 2018), which is based on Scrum and Extreme Manufacturing, is one the most complete framework which could be used in discrete manufacturing or batch production. According the Scrum Inc. organisations using Agile practices include Saab, Boeing, John Deere and Volvo. Most likely low adoption rate is related to Lean management and Agile Manufacturing having significant footprint and Agile software development is perceived more focused on developing highly customised, one-off solutions throughout software lifecycle. Quite like highly customised products, which are constantly and rapidly evolving based on changed needs, resulting really short production batches.

Based on this initial information, a decision was made to survey current situation by interviewing team members contributing to product development process, whether assumed need for change is present and what are most pressing issues. If the survey indicated requirement for a change, a literature study, evaluation and suggestion of the most feasible methodologies and practices would be done. As DMA representatives wanted me to also support in experimentation, an implementation plan and fulfilment of that plan was added into the research scope. This of course prolonged the schedule several months but gave me an opportunity to measure real-life results by having web-based, structured surveys before and after experimentation.

As there is still quite small amount of research about implementation Agile methodologies outside ICT industry. This research and its results can prove to be of some value in future meta-studies. As DMA's industry is specific and even globally quite niche, it is not likely to be replicated in scale. But if it increases Agile methodologies awareness and encourages readers to interdisciplinary thinking, this research has earned its existence.

2 AGILE (SOFTWARE) DEVELOPMENT

Agile Software Development (in short Agile) is a philosophy, an umbrella term for a set of methods and practices based on the values and principles expressed in the Agile Manifesto, where solutions evolve through collaboration between self-organizing, cross-functional teams utilizing the appropriate practices for their context. (Agile Alliance 2018.)

Agile has a lot of overlap with Lean management such as continuous improvement, making things visible, empowerment of teams, process flow, value, value streams and removal of waste. Both Agile and Lean fall under study of Systems Thinking. There are probably hundreds of methodology variants that follow Agile values and principles. Each methodology prescribing distinct set of practices and tools.

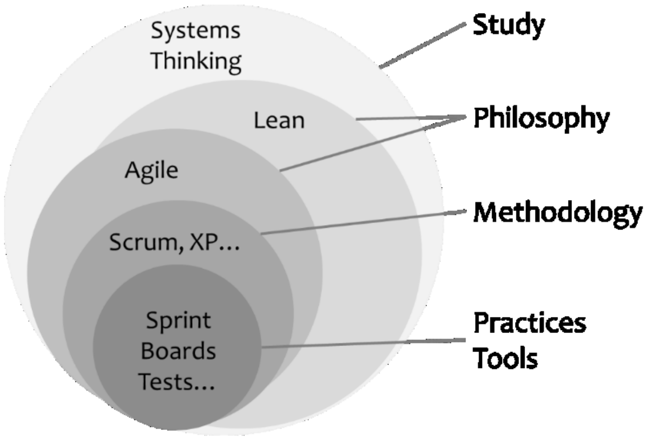


Figure 1. Positioning of Agile Software Development.

Most distinctive difference between Agile and previously more used Waterfall methodologies is iterative and incremental approach. Instead of trying to predict the outcome into smallest detail early in the Waterfall project and delivering everything once, Agile delivers in iterations which provide new, changed or fixed functionality in increments throughout the project.

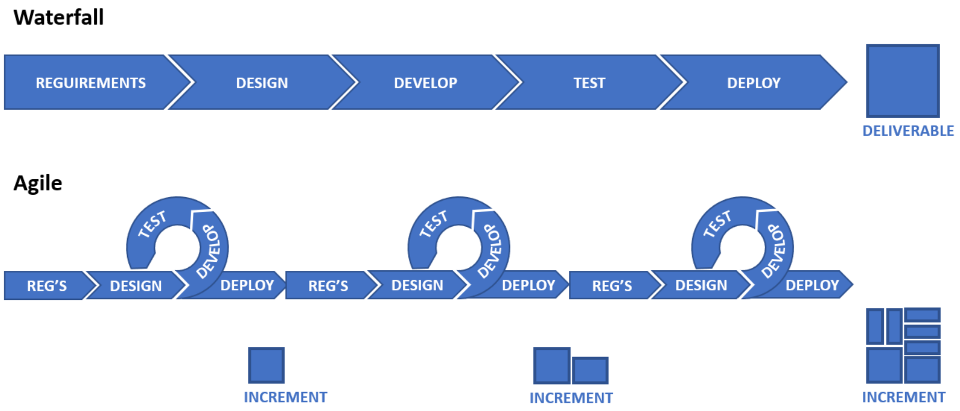


Figure 2. Difference between Waterfall and Agile approach.

Although Agile Software Development and Iterative and Incremental Development (IID) are considered to be quite modern their evolution dates back to 1930's to work of Walter Shewhart, who proposed using a series of short “plan-do-check-act” (PDCA) cycles for quality improvement. PDCA was also promoted since the 1940s by W. Edwards Deming what he eventually described in his book Out of Crisis in 1982. Although he thought “plan-do-study-act” (PDSA) would better elicit the importance of learning from the experiment. Earliest successful iterative software development projects can be tracked to 1957 when IBM’s Service Bureau Corporation used technique resembling many of the Extreme Programming practices. While there is some evidence and publications of use of IID from 1960s to 1980s and grim reviews pointing out strict, document-driven, single-pass waterfall approaches inability to deliver quality software, it was not before 1990s when IID started to gain real traction. During 1990s public awareness of IID in software development raised due to hundreds of publications. This also led into birth of dozens of IID methods like Rapid Application Development (RAD), Dynamic Systems Development Method (DSDM), Scrum, Rational Unified Process (RUP), Experimental Programming (XP) and Feature-Driven Development (FDD). (Larman & Basili 2003, 47–48.)

In 2001, group of 17 software development process experts including Kent Beck (XP/TDD), Jeff Sutherland (Scrum), Ken Schwaber (Scrum), Ward Cunningham (XP/Wiki), Jim Highsmith (ASD) and Alistair Cockburn (Crystal) authored an Agile Manifesto promoting modern, simple IID methods and principles. At the same occasion, Agile Alliance was formed. Agile Manifesto defines Agile philosophy by four values:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan.

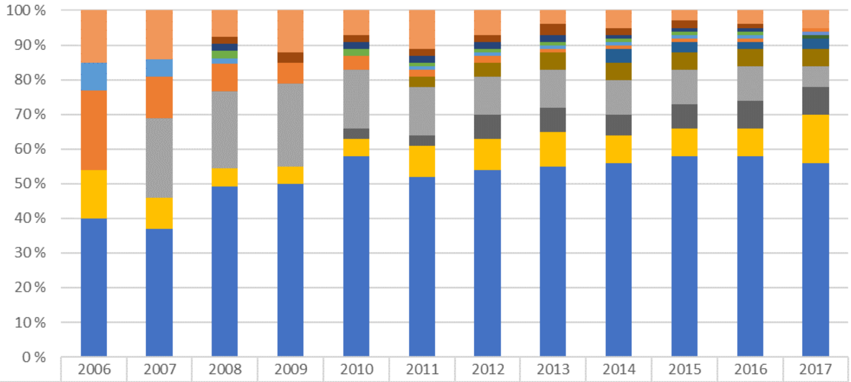
And twelve principles (agilemanifesto.org, 2018):

- Customer satisfaction by early and continuous delivery of valuable software
- Welcome changing requirements, even in late development
- Working software is delivered frequently (weeks rather than months)
- Close, daily cooperation between businesspeople and developers
- Projects are built around motivated individuals, who should be trusted
- Face-to-face conversation is the best form of communication (co-location)
- Working software is the primary measure of progress
- Sustainable development, able to maintain a constant pace
- Continuous attention to technical excellence and good design
- Simplicity-the art of maximizing the amount of work not done-is essential

- – Best architectures, requirements, and designs emerge from self-organizing teams
- – Regularly, the team reflects on how to become more effective, and adjusts accordingly

Today there are tens of Agile methodologies and if you count all variations, fusions, hybrids and flavours there are hundreds known variants of Agile. To evaluate which Agile methodology would be most suitable framework in this case, population was restricted to most used frameworks and the ones which popularity have increased most during past couple of years.

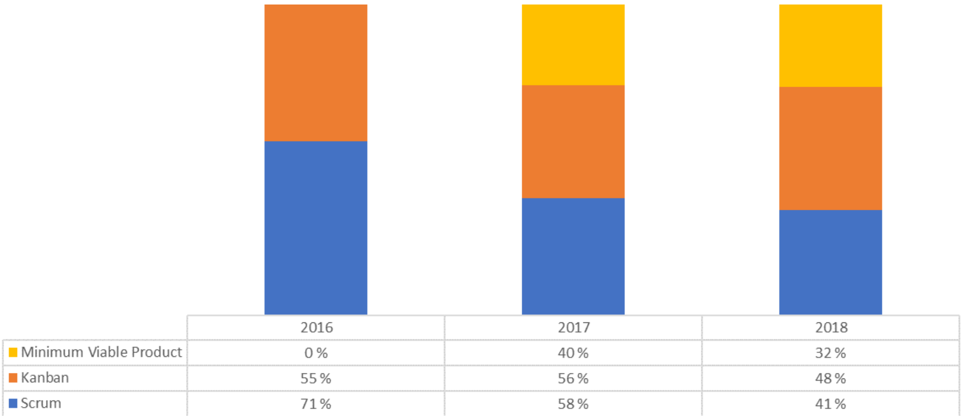
VersionOne has done their State of Agile-Survey now for 12 years, which includes information about popularity of different methodologies. During the time this survey has been made, Scrum has always been most popular Agile methodology followed by the organisations responding to survey. Combined popularity of Scrum, Experimental Programming and hybrid of these two has been between 70% – 80%. XP's usage has significantly dropped during past ten years and it seems many XP practices has become software development best practices which are being followed despite the framework being used. Early surveys show some use of DSDM, AUP, FDD and LSD, which all have now fallen into Others category. Those has been replaced by Kanban, Scrumban and Iterative Development. Latest newcomers to list are Lean Startup and Spotify Method. (VersionOne 2018.)



	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Other	15 %	14 %	8 %	12 %	7 %	11 %	7 %	4 %	5 %	3 %	4 %	5 %
Lean Development			2 %	3 %	2 %	2 %	2 %	3 %	2 %	2 %	1 %	
FDD			2 %		2 %	2 %	2 %	2 %	1 %	1 %	1 %	
AUP			2 %		2 %	1 %	1 %	1 %	1 %	1 %	1 %	
DSDM	8 %	5 %	1 %			1 %	1 %	1 %	1 %	1 %	1 %	
XP	23 %	12 %	8 %	6 %	4 %	2 %	2 %	1 %	1 %	1 %	1 %	1 %
Lean Startup												1 %
Spotify model												1 %
Iterative development									4 %	3 %	2 %	3 %
Kanban						3 %	4 %	5 %	5 %	5 %	5 %	5 %
Scrum/XP Hybrid		23 %	22 %	24 %	17 %	14 %	11 %	11 %	10 %	10 %	10 %	6 %
Scrumban					3 %	3 %	7 %	7 %	6 %	7 %	8 %	8 %
Custom Hybrid	14 %	9 %	5 %	5 %	5 %	9 %	9 %	10 %	8 %	8 %	8 %	14 %
Scrum	40 %	37 %	49 %	50 %	58 %	52 %	54 %	55 %	56 %	58 %	58 %	56 %

Table 1. Agile methodologies use globally (VersionOne 2006–2018).

Codento has done similar assessment in the Finnish market. Most significant difference is that Codento reports methodologies and practices as concepts on the same level. Based on this survey Scrum was in 2016 still most popular methodology with 71% of organisations using it. Kanban being second with 55,4% (Aukia, Luoto & Tiainen 2017). In two years, this has been changed in favour of Kanban having 48% of organisations following it and Scrum going down to 41% (Aukia, Iivonen & Luoto 2018). Most likely high share of Kanban compared to VersionOne report is related to popularity of kanban board as a tool, not Kanban as an Agile methodology. It is also good to note that 32% of organisations use Minimum Viable Product which is a practice used in Lean Startup methodology.



	2016	2017	2018
🟨 Minimum Viable Product	0 %	40 %	32 %
🟧 Kanban	55 %	56 %	48 %
🟦 Scrum	71 %	58 %	41 %

Table 2. Agile methodologies use in Finland (Aukia et al. 2017–2018).

Additionally, from the research subject and manufacturing industry point of view, one interesting framework is Extreme Manufacturing (XM) used by Wikispeed Ltd to build 100-miles-per-gallon vehicle, which has evolved to Scrum for Hardware during last couple of years and has been adopted by some companies in automotive and aviation industry (Brown & Justice, 2018).

2.1 Scrum

According to VersionOne State of Agile report Scrum has been the most used agile methodology throughout past decade. It was authored by Jeff Sutherland and Ken Schwaber at 1995 by inspiration of article called “The New New Product Development Game” from Takeuhci and Nonaka (1986). It suggests that old sequential development approach should be replaced with new approach with heavily overlapping development phases providing faster lead time and more flexibility. Sutherland suggests that Scrum provides 300–400 percent productivity improvement and doubling of quality compared to traditional approaches. (Sutherland 2015, 31 – 34.)

Scrum framework is medium prescriptive having three roles, five events and three artefacts as follows:

- Development Team
- Product Owner — Roles
- Scrum Master
- Sprint
- Sprint Planning
- Daily Scrum — Events
- Sprint Review
- Sprint Retrospective
- Product Backlog
- Sprint Backlog — Artefacts
- Increment

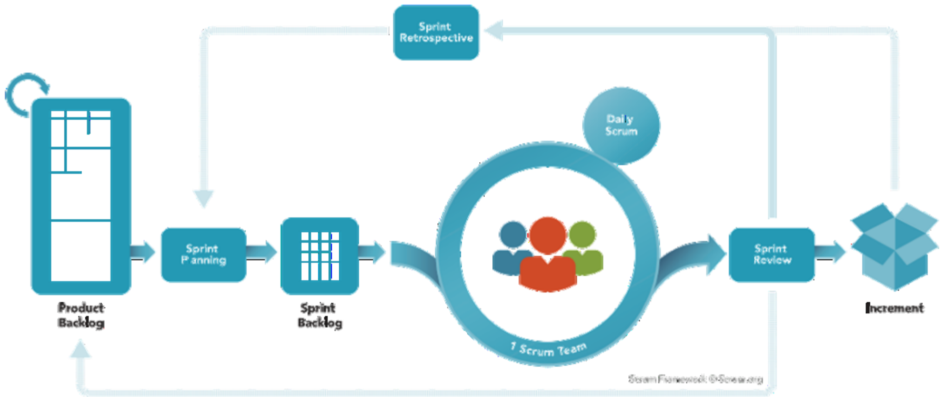


Figure 3. Illustration of Scrum framework (Scrum.org n.d.).

Scrum teams are cross-functional having all necessary skills needed to complete the increment, the outcome done in the sprint. Team individuals may have specialized skills and focus, but whole team is accountable of the results. Team size should be rather small, around 7–8 persons ± 2 preferably working in same premises. Having small team size in same location makes communication between team members easier. Teams are self-organising and self-managing, empowered to make decisions about how assignments are resolved. This autonomy is underpinned by sense of purpose above ordinary. (Sutherland 2015, 41–45 & 58–61.)

Product Owner is responsible of Product Backlog consisting all the requirements for any changes waiting for accomplishment in priority order. Inspiration to Product Owner role came from Toyota Product Development System's Chief Engineer role, who typically are senior engineers having wide experience and knowledge about domain team is working on but also having good understanding of customer needs and market situation. Product owner need to have constant dialog with the team to make sure that planned value is realized when increment is released. (Sutherland 2015, 176–180.)

Scrum Master is responsible for promoting Scrum practices, supporting team members to work accordingly and process being effective. Scrum Master is like a Project Manager in waterfall project, but with more servant-leader approach. This person is responsible of facilitating all the Scrum events, making sure that there is required transparency and most importantly, helping team to remove impediments hindering team to finish requirements. Scum Master has also significant role to make sure that process team follows is continuously improved. (Sutherland 2015, 61–62.)

Sprint is a one to four-week time-box, in which potentially releasable product increment is created. Sprint length should remain unaltered between Sprints. One sprint consists of Sprint Planning, Daily Scrums, development work, Sprint Review and Sprint Retrospective. During the Sprint you should not assign new tasks to team or reprioritize items in that Sprint, but task scope can be clarified between the team and the Product Owner. Usually teams use simple boards to illustrate state of different tasks (Figure 4). Even though Scrum does not limit number of tasks assigned to one person, it is advised to have as few simultaneous tasks as possible to avoid waste related to task switching. Team also should concentrate on finishing as many shippable products as possible if it seems like they are not able to reach Sprint objectives. Scrum's progress is usually followed with burndown chart illustrating current estimation of remaining work in on going Sprint (Figure 5). (Sutherland 2015, 72–76 & 88–94.)

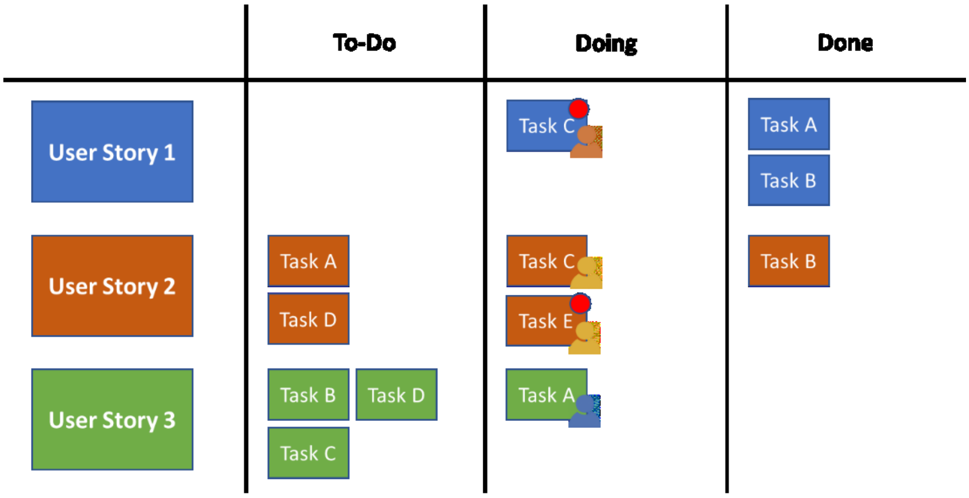


Figure 4. Example of Scrum Board.

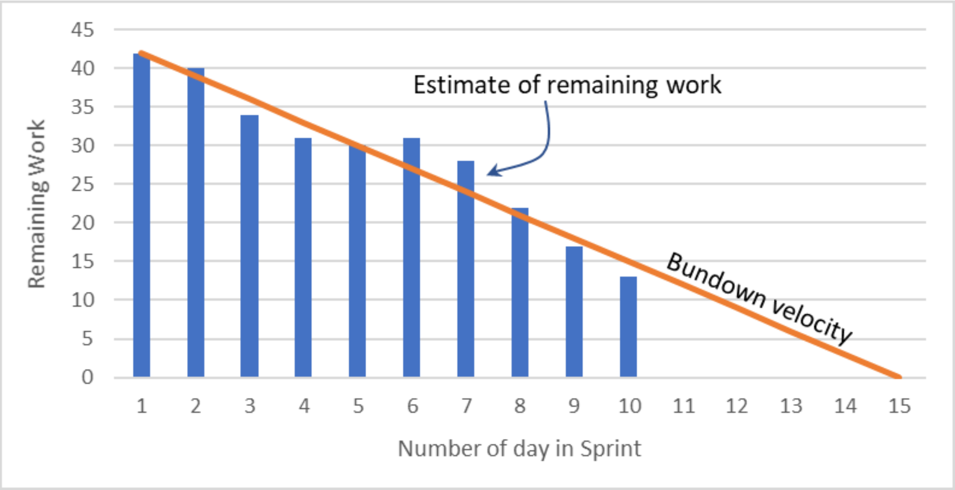


Figure 5. Example of a Scrum Burndown Chart for a three week sprint.

Content of each Sprint is planned in Sprint Planning and is documented into a Sprint Backlog. Participants for Sprint Planning are Development Team, Scrum Master and Product Owner. Based on prioritised Product Backlog, team selects Stories (requests) which can be accomplished during upcoming Sprint and how work is done. At the same time knowledge about purpose of each Story is transferred from Product owner to Development team to form common understanding of Definition of Done for the Increment. According to The Scrum Guide (Sutherland & Schwaber 2017, 10–11), Sprint planning should take no more than eight hours depending length of the Sprint. (Sutherland 2015, 138 & 235–236.)

During the Sprint, Development Team will hold a daily 15-minute meeting called Daily Scrum. During the meeting team gathers around Scum Board and each Team member has couple of minutes to answer three questions:

- What did you do yesterday to help the team to finish the Sprint?
- What will you do today to help the team to finish the Sprint?
- What obstacles are getting in the team’s way?

Rationale behind this meeting is to boost communication between team members and make possible impediments visible. This also makes sure that every team member is aware of Sprint’s progress. Usually raising issues in Daily Scrum leads into after meeting discussion or swarming with team members able to help removing the impediments. (Sutherland 2015, 76–79.)

A Sprint Review is held at the end of each Sprint and it should accommodate anyone wanting to participate. This event should take no more than 4 hours to one-month Sprints. In this event, team should demonstrate all features which meets the Definition of Done, are completely finished and able to release into production. Product Owner will show current state of Backlog and there is a discussion providing input for subsequent Sprint Planning. (Sutherland & Schwaber 2017, 13.)

Sprint Retrospective is the key event to make sure there is continuous improvement on followed process. In this meeting Scrum Master and the Team will go through the last Sprint. Did they follow the agreed Scrum process? What were the impediments during the Sprint and how they could avoid those in the future? Based on these discussions team will pick one or couple of improvement tasks for the next sprint, which will be regarded like any other Story to be accomplished. Sutherland (2015, 145–160) greatly emphasizes team member happiness and its impact on productivity. Sprint Retrospective is seen as a valuable opportunity to monitor and increase happiness. (Sutherland & Schwaber 2017, 14.)

Product Backlog is a list of all the new features, functions, fixes and enhancements called stories to change the current product. The Product owner is responsible of all the Backlog content and prioritisation. Backlog refinement is an activity where Product Owner together with the Development Team update details, estimates and order of assignments during the Sprints. Refinement takes around 10% of Development team capacity. (Sutherland & Schwaber 2017, 15.)

The Increment is a combination of the Product Backlog items which meet the Scrum Team’s Definition of Done criteria composed during Sprint Planning. All the Stories within an Increment are potentially Customer shippable and finished. Not necessary product but part of the product as a function providing value to users of the product. It is up to Product Owner and release scheduling whether the Increment is released immediately or later. (Sutherland & Schwaber 2017, 17.)

2.2 Extreme Programming

Extreme Programming (XP) was authored by Kent Beck at 1999 focusing to produce high-quality software in short iterations. It was developed based on Beck’s work on Chrysler’s payroll system implementation during 1996–1999. Based on Agile surveys (VersionOne 2018) use of XP as a methodology has declined from 2005 when it was used by 30–40% of organizations as standalone framework or in combination with Scrum, close to 7% and its standalone use has almost vanished. At the same time many of the XP principles and practices has been adopted by other Agile methodologies or those has become software industry best practices. (Beck 2005, 125–129.)

Extreme Programming has thirteen primary practices, which have evolved since its initial release. These thirteen practices are (Beck 2005):

- Sit Together
- Whole Team
- Informative Workspace
- Energized Work

- Pair Programming
- Stories
- Weekly Cycle
- Quarterly Cycle
- Slack
- Ten-minute Build
- Continuous Integration
- Test-first Programming
- Incremental Design

XP teams are prescribed to Sit Together while doing development work in common open space. This is seen to have positive effect to intra-team communication and build fellowship. It also reduces significantly required walking or travelling time which can be used to contribute to customer value. This practice fits together with Whole Team and Informative Workspace principles. Principle of Sit Together doesn't exclude use of XP in geographically dispersed teams, which is usually the case in large organisations where different expert communities are in separate sites or Offshore resources are being used. In such case more attention should be given to ensure feedback by for example having several planning meetings during the week. (Beck 2005, 37–38 & 149–150.)

Whole Team advocates for cross-functional teams having all the necessary skills to proceed successfully in the project including if possible, a customer representative to contribute and elaborate feature stories. Principle goes beyond just cross-functional team as team members should feel belonging to a team, support each other and omit shared responsibility of results. XP sets the ideal team size based on Gladwell's (2000, 175–181) description of discontinuities. Teams should be split if they have more than 12 members as "Twelve is a number of people who can comfortably interact with each other in a day". To scale, team of teams can be used. Generally, instead of making teams bigger, splitting problems into smaller problems is advised. Having non-fulltime assignments to team should be avoided as task switching will increase waste, having several objectives blurs focus and decreases feeling of belonging to a team. (Beck 2005, 38–39 & 111–112.)

Informative Workspace principle describes a story card wall which should be put in central place in a team workspace. This card wall (Figure 6) provides a quick glance on what team is working on and what kind of things are in the pipeline. Card wall helps to pinpoint if team is having issues in planning, estimation or execution. Additionally, charts can be used to visualise progress in long-term projects. Workspace should support Pair Programming practice by having two-person workstations and informal information exchange between team members, simultaneously providing ability to work privately if needed in separate cubes. (Beck 2005, 40–41.)

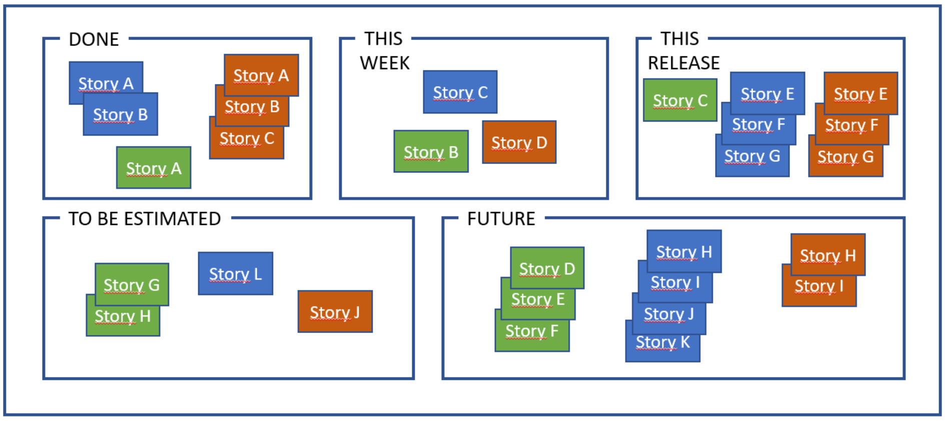


Figure 6. Example of story card wall used in Extreme Programming.

Energized Work practice follows sustainable pace Agile principle and is an evolution of “40-hour week”. People should work only as long as they can sustain productive pace. Working significantly longer than normal working hours should be considered more of losing control than sign of commitment as productivity sinks rapidly after eight hours of focused working time usually leading into value shrinkage. Same applies to working while being sick. Concentrating on rapid recovery will have more positive impact than working ill. This also protects rest of the team losing more productivity because of illness. During active working time focused contribution should be protected against external disruptions by shutting off phones and other communication channels for unsolicited contacts. (Beck 2005, 41–42.)

Pair Programming is a practice where two people are sitting at one computer. One person is programming while pair having a constant dialog of how things should be resolved and making sure that there are no defects injected into the code. This practice promotes knowledge sharing and reduces significantly quality issues and quicker remedy of possible issues. Pair should switch roles during pair programming session and there should be rotation in pair members so that people work with everybody in the team regularly. Pair programming should result better or close to same volume of code than done by itself, but a lot lower amount of errors. (Beck 2005, 42–43.)

Stories refer to assignments, tasks, feature requests or requirements similarly as in many Agile methodologies. Stories should be written according to customer-visible functionality for better general comprehension of requests better and have a short, descriptive name. Estimations should be done as soon as stories are written to give possibility to early reprioritization of potential ideas. Physical cards on wall are preferred over virtual systems because of their ease of use and practicality. (Beck 2005, 44–45.)

Work in XP is planned in Weekly and Quarterly Cycles. Beginning of each week, there should be a planning meeting reviewing last week's progress against expected results, having customer picking on-going weeks stories for implementation and breaking these stories into tasks which will be signed up by team members. Planning is considered as a necessary waste and such time consumed on planning should be minimized. When team's experience increase, planning should in optimum only take only an hour a week. After every week all assigned tasks should pass testing and there should be another deployable version of software. Once a quarter team should reflect the whole project, progress and its alignment with broader goals. Team should identify issues and bottlenecks and decide how to resolve those, plan the theme and initially pick theme-related stories to be implemented. Quarterly cycles are also good for longer term improvement experiments. (Beck 2005, 47–48.)

Plans made in Weekly and Quarterly cycles should always include tasks with minor importance which can be dropped if results begin to drag. This kind of Slack combined with clear and honest stakeholder communication improves credibility. And being able to release according to commitments increase trust between team and other stakeholders. (Beck 2005, 48.)

Team should aim at being able to build the whole system and run all the tests in ten minutes. This requires all the build and test tasks to be automated. Also, this ability to build and test whole environment should be used frequently to verify new code put into repository. Similarly, new code should be merged into a single, common main branch several times a day to test possible interoperability issues with rest of the code base. Usually automated code builds and tests are made asynchronously after code commit providing near real-time feedback. This can be combined in the end of each Pair Programming session so that possible issues can be immediately fixed and avoid waste resulting from task-switching. Continuous Integration (CI) and Continuous Delivery or Deployment (CD) are practices followed by most, if not all, modern application development teams. (Beck 2005, 49–50.)

After weekly planning session, development should begin with writing tests which will be run after story is completed. Test-first Programming practice focuses development to features requested. If additional features arise during development, those should have their own tests written after minimum deployable feature has been finished. Being unable to write a test usually indicates design problems in code. Refactoring the code and making it simpler should help. Having tests for every new feature demonstrates code cohesiveness and increases trust between team members. There is on-going development done to achieve continuous testing during development, but these are still running too slowly for real-time development. (Beck 2005, 50–51.)

Last principle is Incremental Design, which was called in earlier version of XP Refactoring. Incremental design suggests investing in to the design of the system every day. As best possible design for the system evolves all the time, development work to achieve that goal should be gradual and done while making story changes in respective code areas. Considering continuous development of technology, designing as early as possible in start of the project, should be postponed close to when design is required. This is also a form of risk management as there is only minimum investment on long-term design and that cost is divided on a long period of time. Beck also suggest that many design issues are such that having experience how to solve them doesn't even exist yet. In those situations, incremental approach gaining the experience while implementing new features is the most preferable approach anyway. (Beck 2005, 51–53 & 103–109.)

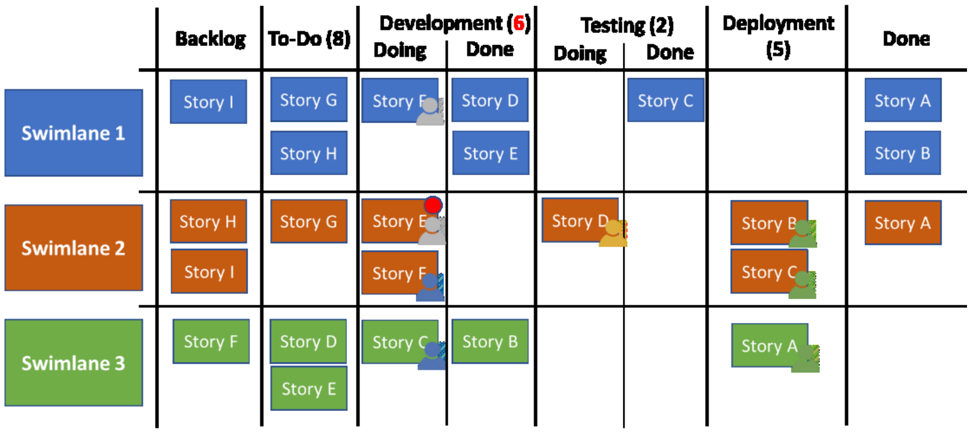
2.3 Kanban

Kanban is an agile method which evolution started in Microsoft 2004 when first virtual kanban board was implemented. The evolutionary, incremental process improvement methodology Kanban emerged during 2006 and 2008 at Corbis by David J Anderson. Kanban is a pull system underpinned by Toyota Production System (Lean) alternative to Eliyahu Goldratt's Drum-Buffer-Rope application of the Theory of Constraints. The Kanban Method started to grow in community adoption around 2007 and has been evolving in wider Lean software development community during the years. (Anderson 2010, 3–8.)

Kanban has five core practices to achieve emergent set of Lean behaviours in organisations:

- Visualize workflow
- Limit work-in-progress
- Measure and manage flow
- Make process policies explicit
- Use model to recognize improvement opportunities

Visualization of the workflow is achieved by use of card walls (kanban board) illustrating different process phases in columns and different products or work item types as separate rows called swim lanes (Figure 7). Each task in the system has its own card indicating basic information like unique task id, descriptive name of task, date of assignment and fixed delivery date if such exist. Card colour usually indicates a certain product or assignment class of service. Assignees of the task are usually visualized by magnetic avatars on top of the card or by writing a name on a board on top of the card. Additionally, stickers can be used to indicate different conditions like task being blocked. (Anderson 2010, 64–71.)



	Backlog	To-Do (8)	Development (6)		Testing (2)		Deployment (5)	Done
			Doing	Done	Doing	Done		
Swimlane 1	Story I	Story G Story H	Story F	Story D Story E		Story C		Story A Story B
Swimlane 2	Story H Story I	Story G	Story E Story F		Story D		Story B Story C	Story A
Swimlane 3	Story F	Story D Story E	Story C	Story B			Story A	

Figure 7. Example of kanban board.

Kanban aims to significantly increase task lead time by limiting work-in-progress team members are allowed to pull from previous phase of process. Number of tasks allowed in certain point of process is indicated by a figure in column title. Based on queuing theory called Little's Law there is linear relationship between quantity of work-in-progress and average lead time. Of course, after certain point this will also affect to velocity (production volume) negatively and team should experiment with WIP limits to discover optimal balance. Even early on implementation, it is advised to have in maximum three open tasks per team member to avoid multitasking and loss of active working time caused by context switching. Having WIP limits smaller than team size will naturally lead into pair programming practice. (Anderson 2010, 25–28 & 113–122.)

As such Kanban doesn't rule use of any specific reporting even though it prescribes measurement and management of the flow. Best practice report to visualize lead time and number of work-in-progress over time is a Cumulative Flow diagram (Figure 8). Addition to reporting Average Lead Time, Lead Time spectral analysis should be used to evaluate how predictable Average Lead Time is. If Fixed Delivery Date class of service is used, Due Date performance should also be reported to evaluate how well agreed schedules are being met. (Anderson 2010, 139–147.)

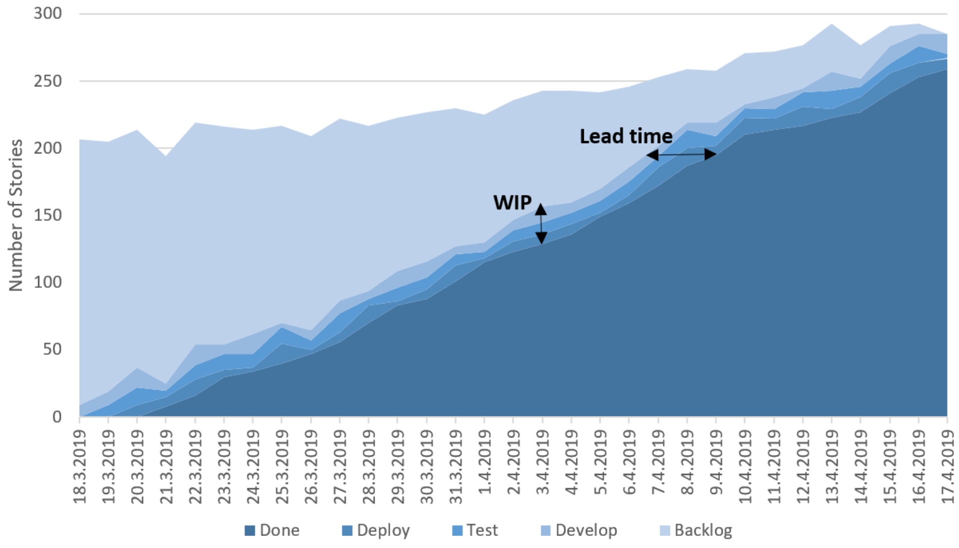


Figure 8. Example of Cumulative Flow Diagram (Anderson, 2010).

Even though Kanban is inclined to highlight Lead time, it is a good practice to follow throughput with velocity metric indicating how many task, stories or story points were completed within certain period and flow efficiency calculating ratio of active work time on task to lead time. This metric visualizes amount of time task spend in process buffers or queues waiting for additional value creation. As Kanban focuses on constant evolutionary change, often referred as Kaizen, it is a good practice to follow number of blocked tasks relative to time and how process issues get fixed (Figure 9). Finally tracking number of bugs or product defects over time helps organisation to understand how the quality resulting from the process is being improved. (Anderson 2010, 139–147.)

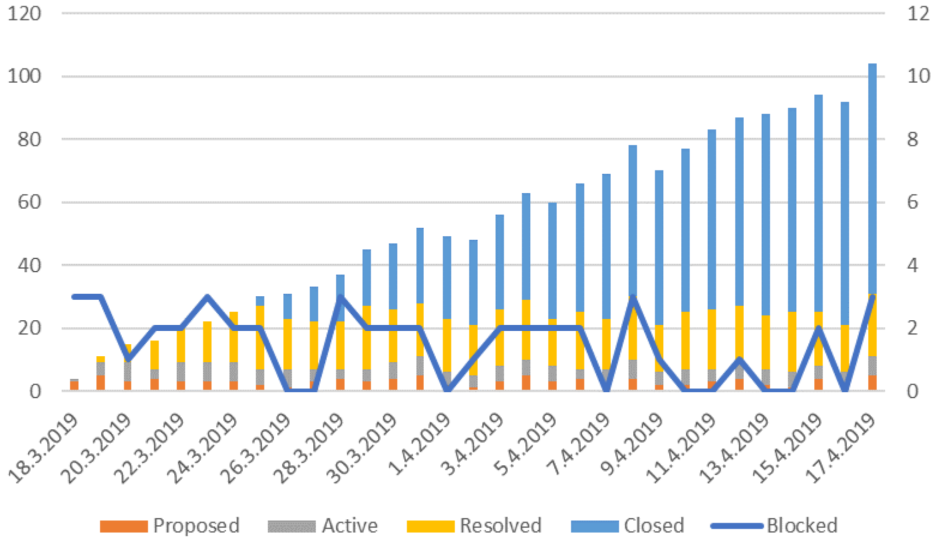


Figure 9. Example of Issues and Blocked items chart (Anderson, 2010).

As Kanban is relatively non-prescriptive methodology, it requires significant amount of additional context specific policies. Policies should be agreed by team members and stakeholders and they need to be explicit. Usually some of the generic practices like Definition of done, Daily Standups, After Meetings, Backlog prioritization, Release planning, Operations reviews and Issue management are encouraged to be defined. Definition of done defines when task is possible to move to next phase in the process. Daily Standups are short daily meetings concentrating on blocks preventing team members to finish tasks. Blocks are not typically solved during Daily Standups, but they are handled in some type of swarming activity called "After meeting" or “Parking Lot Discussion”. Backlog prioritization is an activity where stakeholders or business owners decide which work items will be moved from Backlog to To-do phase as open slots emerge when previous tasks move further on the process. Release planning is required if Kanban teams are required to participate release actions of finished product components or if release activities are also managed with Kanban board. Issue management defines how blocked tasks are handled, documented and further on avoided in the future. This usually includes some sort of escalation if team cannot resolve issue on its own and help from other departments or 3rd parties are required. Operations reviews should be held once a month or bi-monthly including all members actively participating to end-to-end process and stakeholders providing input or consuming outputs of process. This meeting concentrates on result achievements, optimization of the process and how issues can be avoided in the future. (Anderson 2010, 82–88 & 159–165 & 235–239.)

Anderson suggests that there are three types of primary improvement opportunities which should be concentrated on. First one is identifying and removing process bottlenecks with “Five Focusing Steps”, a continuous improvement framework from The Theory of Constraints developed by Eli Goldratt in 1984. Framework states:

1. Identify the constraint.
2. Decide how to exploit the constraint.
3. Subordinate everything else in the system to the decision
4. Elevate the constraint.
5. Avoid inertia; identify the next constraint and return to step 2

Second improvement opportunity is Lean/TPS based elimination of waste. In Andersson’s approach waste can be classified into three main categories: transaction costs, coordination costs and failure load. Transaction costs are generated as setup and delivery costs of new products and those should be balanced based on benefits of new release and effort required. Coordination costs are related to assigning people to tasks, scheduling events or coordinating work of two or more people toward a common goal. These should be minimized. Failure load is a work generated because of some earlier deliveries failing, such as product defects, poor design, increased amount of service requests etc. This type of waste uses capacity, which could be used for creating new value-adding features and should be avoided by pursuing high product quality. Third area of improvement is reduction of variability, which is addressed in Kanban with Statistical Process Control (SPC) pioneered by Walter Shewhart in 1920s and further developed by W. Edwards Deming in his Theory of Constraints. Both Lean/TPS and Six Sigma has adopted SPC to improve process flow. SPC divides sources of variation into two categories. First category is Internal variations, which are also referred to as change-cause variations, like work item size, type, class of service, irregular flow and rework. These can be controlled using policies that define product development lifecycle and project management process in use. Second category is External variations, which are also referred as assignable–cause variations, like requirements ambiguity, expedite requests, irregular flow, market factors, staffing factors and challenges in scheduling coordination. These can be managed using issue-management and resolution capabilities and reduced by root-cause analysis and elimination capabilities. (Anderson 2010, 187–192 & 232.)

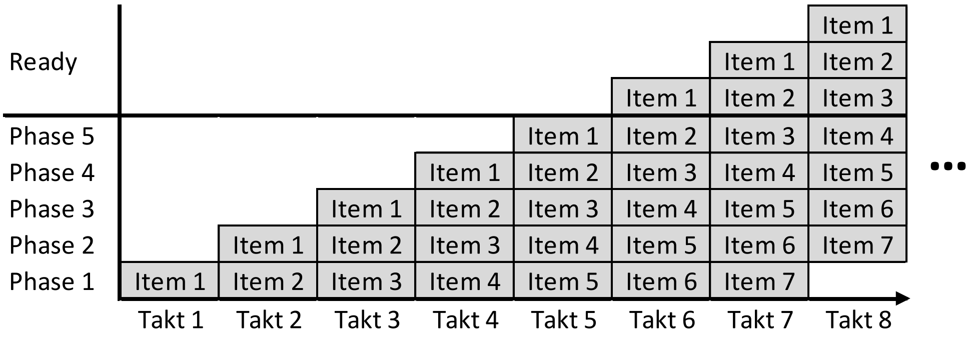
As a result of successful Kanban implementation following organizational behaviour are emerged (Anderson 2010, 16):

1. Process uniquely tailored to each project/value stream
2. Iterationless development
3. Work scheduled by Cost of delay
4. Value optimized with Classes of service
5. Risk managed with Capacity allocation
6. Tolerance for process experimentation
7. Quantitative management
8. Viral spread of Kanban across the organisation
9. Small teams merged for more liquid labour pools

2.4 Scrumban

Scrumban is a hybrid of Scrum and Kanban methodologies authored by Corey Ladas at 2008. Even though it is considered as a separate method, Ladas advocates benefits of Kanban over other agile methodologies. Scrumban provides a framework how organisations currently following Scrum framework, may transform to use Kanban. He suggests that if organization is currently following waterfall-type process with functionally aligned teams, it should implement Kanban. But if organization has now more project-aligned teams, then Scrum with cross functional approach could prove to be more convenient. If again organization is already following Scrum they are encouraged to move to Kanban with evolutionary approach. As Scrum is widely adopted and XP has become more of the set of best practices followed in software industry, Ladas describes in detail how team could transform evolutionary way. (Ladas 2008, 82–85.)

Toyota Production System (TPS) has a goal of having only one work item in each workflow phase and work items moving between the phases in takt, which is a time between the start of production phase of two units without need of buffering between workflow phases. Figure 10 illustrates how items move between phases during equally sized time slots (takts) throughout the synchronized workflow. (Ladas 2008, 34–36.)



	Takt 1	Takt 2	Takt 3	Takt 4	Takt 5	Takt 6	Takt 7	Takt 8
Ready						Item 1	Item 1 Item 2	Item 1 Item 2 Item 3
Phase 5					Item 1	Item 2	Item 3	Item 4
Phase 4				Item 1	Item 2	Item 3	Item 4	Item 5
Phase 3			Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Phase 2		Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7
Phase 1	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	

Figure 10. Example of 7 items in 5 phase workflow.

Same rationale applies to moving from waterfall to agile. Instead of having one iteration to achieve 100 requirements Scrum splits requirements to 10 sprints having 10 requirements each. Ladas's testimonial is like in Lean/TPS, that ultimate optimum is to have one-piece flow without iterations at all. But as it is impossible to know development work time required in each workflow phase beforehand, synchronous workflow cannot be implemented. Instead implementing Kanban pull system with minimum buffers to balance flow should be set as a goal. (Ladas 2008, 44–53.)

Scrumban as a framework prefers specialized teams instead of cross-skilled generalist team members with ability to fulfil all tasks end-to-end, which is a goal of a Scrum team. Usually people are highly skilled and motivated in certain tasks but at the same time they perform worse in other tasks. When optimizing resourcing to minimize lead time, preference is to use highest skilled persons to fulfil each task. In this type of resourcing it also makes more sense train persons strong areas of skills, not the weak ones, as it provides significantly improved lead time. (Ladas 2008, 73–79.)

A most distinctive practice suggested in Scrumban is use of a bucket brigade-style of skills combination to eliminate buffers. As an example (Figure 11), it could consist 3 persons, each having 2 high skill areas of totally 4 phases of workflow. First person would do phase 1 and phase 2 tasks, second person would do phase 2 and phase 3 tasks and third person phase 3 and 4 tasks. In this model person 3 would pull immediately work from person 2 which is in phase 3 and respectively person 2 would pull tasks in phase 2 from person 1, who would take the next job from backlog to phase 1. Person 2 would act like a flexible buffer but simultaneously actively contributing to product. (Ladas 2008, 146–151.)

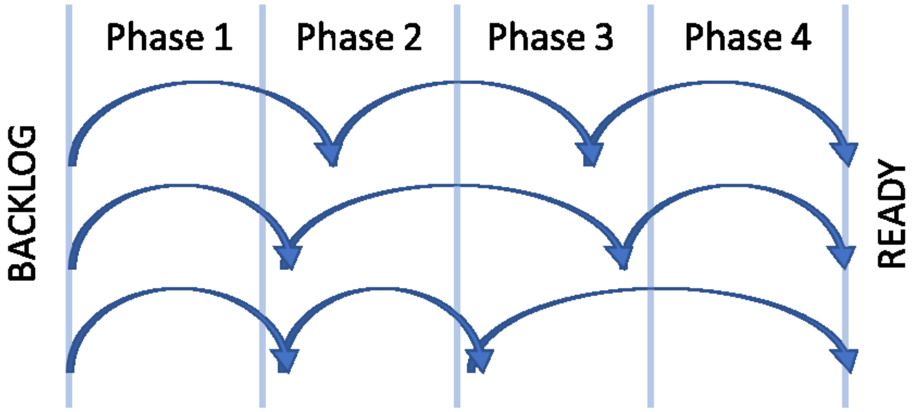


Figure 11. Bucket Brigade-style bufferless flow.

More complex workflows having 6 phases or more could be handled without buffers with 2–3 persons like illustrated in Figure 12 by using return flow and 2 cycles (Ladas 2008, 152–157). This kind of setup seems feasible only when team size is small and complex workflow followed. Otherwise it is contradicting with preference to use of specialized teams.

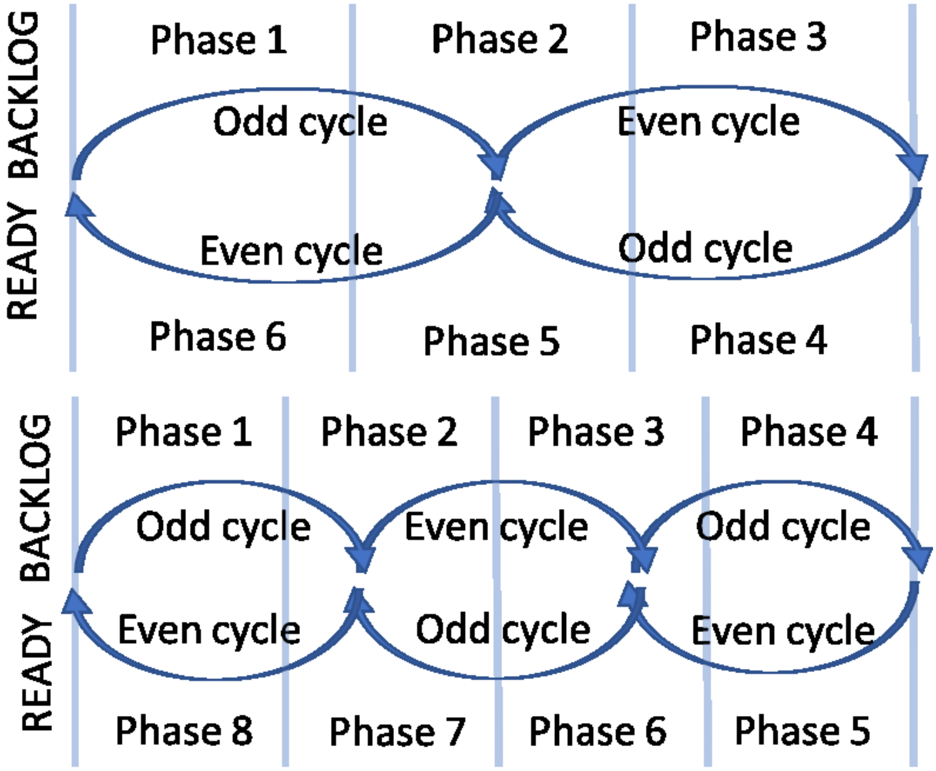


Figure 12. Complex workflow bucked brigade examples.

One of the Scrumban principles is Release Early, release often (RERO), which was initially published in mid-1990’s which is also applied in Extreme Programming and other Agile methodologies. Scrumban emphasizes definition of minimum deployable feature set to determine which features are required for logical completeness. As an example, car without transmission is illogical, but car without heated cup holder is not. Even if your competition has released a product with more features, you shouldn't try to compete with scope, but productivity. Key is to provide incremental features in increasing speed to overcome competition. (Ladas 2008, 158–159.)

Lastly Ladas (2008, 163–172) raises the issue regarding prioritization. It's quite usual that around 80 percent of backlog features fall into priority one class. This makes prioritization diluted as "If everything is high priority, then nothing is". One possible solution is Progressive Priority Filter having columns for backlog, priority 1–3 and done. Each priority column has assignment limits in increasing sequence like geometric (2, 4, 8) or Fibonacci (3, 5, 8). Assignments are pulled from priority 1 and that open slot is filled with a priority 2 assignment, which is further replaced by priority 3 assignment and so forth. If a more important assignment appears in the backlog it can replace any assignment in the board which is then returned to backlog. To prevent assignments to languish in lower priority states without getting promoted and fulfilled every assignment should have an expiration date. Another option is to use Perpetual Multivoting in which a voting committee is selected from stakeholders. Each member has an allocation of votes which she can use to promote any tasks on backlog with as many votes as member has and wants. Votes can be recasted any time. When pull event occurs, top-voted item is selected, and votes are returned into pool for recasting. Oldest items which haven't progressed are removed from backlog.

2.5 Lean Startup

One of the most interesting recent methodologies published in 2011 by Eric Ries is the Lean Startup. When the book was published it went immediately to position number two on New York Times (2011) Best Sellers list and has been allegedly sold over million copies over the years, which is exceptionally good for a business book. Just like Spotify model, it appeared also on latest VersionOne's State of Agile-Survey (2018) and it will be interesting to see can it achieve longer-lasting traction.

The Lean Startup methodology base on five principles:

- Entrepreneurs are everywhere
- Entrepreneurship is management
- Validated learning
- Build-Measure-Learn
- Innovation accounting

Even though entrepreneurs and start-ups are most of the time comprised as small companies working in garages in Lean Startup, start-ups are defined as “a human institution designed to create new products and services under conditions of extreme uncertainty”. In larger companies these intrapreneurs are autonomous teams with separate profit and loss statements, facing similar obstacles while trying to innovate totally new business or revolutionizing corporations current offering to meet evolving customer requirements. Innovation in this context should be understood broadly from new inventions to repurposing existing technology or building a novel business model to fulfil diverse customer needs. (Ries 2011, 25–29.)

The Lean Startup claims that many start–ups fall into “just do it” approach without structured methodology causing many of those experiments to at their best reaching mediocre results or worst leading into purposeless chaos instead of transforming the market. This is usually related to fear of introduction of unnecessary bureaucracy and impeding creativity of highly skilled individuals. At the same time, management practices have leapfrogged in several industries by implementation of Lean and Agile methodologies and practices resulting unprecedented increase in effectiveness and efficiency. Adopting these practices in developing new businesses can provide similarly substantial results. (Ries 2011, 15–24.)

Validated Learning is a Lean Startup concept of how to empirically demonstrate the facts related to business opportunities providing rapidly concrete and accurate real-life results to predefined questions instead of using learning as an afterthought excuse for failure. In addition, providing information about viability of demonstrated feature it may also disclose possible changes required to make product successful. (Ries 2011, 37–38.)

Build-Measure-Learn loop illustrated in Figure 13 is the engine for Validated learning. It starts from identification of hypotheses called leap-of-faith assumptions which a start-up would like to test. Most important assumptions are related to innovations value and growth i.e. does innovation provide value for its target customers and is it possible to build sustainable growth for a product. It is important to define success or failure criteria before experimentation. In the next phase a minimum viable product (MVP) is built with least effort and minimum development time. MVP needs to be published to potential customers and their feedback collected (Ries 2011, 75–78). The Lean Startup introduces a Concierge service and Wizard of Oz testing. In Concierge service a start-up team, instead of immediately building an internet service, provides similar service to strictly limited number of customers personally verifying assumed customer needs and behaviour. This also provides possibility for rich qualitative customer feedback during experimentation (Ries 2011, 99–102). Wizard of Oz testing is somewhat similar, but it has intended user interface available, but all or most of the provided functionalities are achieved by manual labour (Ries 2011, 106). After building a product for experimentation and releasing to customer use, measurements are collected forming a data for learning and decision making.

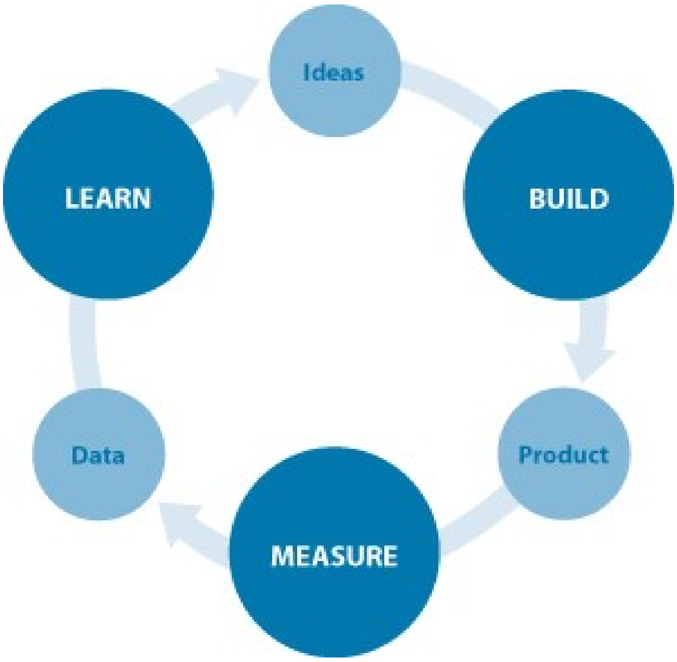


Figure 13. Build-Measure-Learn loop (Ries 2011, 75).

Innovation accounting is a quantitative approach to evaluate results from experiments forming a learning milestone. Learning milestones help start-ups to assess their progress accurately and objectively. Learning milestones are also imperative information to stakeholders to monitor development (Ries 2011, 77). There is always a danger of using vanity metrics, which gives much pleasing picture of progress, but which can be achieved by secondary activities like excessive marketing or dumping product to market with loss-making price. More actionable metrics can be achieved by use of cohort analysis or split tests. In cohort analysis, instead of looking cumulative totals or gross numbers start-up measure each test group using product independently by using relative figures presenting user behaviour (Ries 2011, 123). In split-testing a new and old product or feature is offered to users simultaneously and measurements related to user preference is collected real-time (Ries 2011, 136–137).

After the learning phase a decision to persevere or pivot is done. Idea of the build-measure-learn loop is to answer the question: “are we making sufficient progress to believe that our original strategic hypothesis is correct, or do we need to make a major change?” If change is required, that’s called a pivot, which is designed to test a new fundamental hypothesis about the product, business model or engine of growth (Ries 2011, 149–150). Ries (2011, 172–176) lists ten types of pivots. In Zoom-in pivot, a previously single feature of the product becomes the whole product. In Zoom-out pivot current product becomes one feature or feature-bundle of the new product. Customer segment pivot is required if experimentation shows that product solves real problems but not for the type of customers originally planned. Customer need pivot usually comes to prominence during customer interviews or observations when original product doesn’t solve customer problems but makes related visible. A Platform pivot leads into application turning to a platform or vice versa. In Business architecture pivot mostly high margin, low volume business-to-business product is transformed to a mostly low margin, high volume consumer product, or vice versa. When Value capture pivot is required, a monetization or revenue model of the product is changed. Engine of growth pivot suggest that organizations have three primary engines of growth: the sticky, the viral and paid growth models. In sticky growth model a product is designed for recurrent use and companies try to achieve such customer loyalty that users come back and use service repeatedly. In Viral growth model a word-of-mouth is spreading rapidly and even though customers are not intentionally evangelizing the product, epidemic-like growth is built-in to the product in. This is how most social media services attract increasing number of users. In paid growth model customer acquisition is done mainly by sales and marketing efforts. Every acquisition has its cost and while this cost is lower than additional income, a company can grow profitably (Ries 2011, 209–219). In Channel pivot a company changes the way how it delivers the product to customers. Traditionally options have been through direct sales channel or partner distribution channel. Last option is a Technology pivot, where company discovers a way to achieve same results or functionality by using different technology. This is usually done by established companies to increase product profitability of mature products and markets.

Ries (2011, 111) claims that company which has the shortest time completing the loop is the one learning about the customers and markets fastest and having highest probability to become successful. The Lean Startup also embraces several practices like genchi gembutsu, the five whys and small batch sizes introduced in The Toyota Production Model (Liker 2010, 223–235; 252–254; 21–22). Ries (2011, 255–256) also raises importance of rapid decision making and single authority referring to a Chief Engineer role in The Toyota Product Development System (Morgan & Liker 2006, 117–120).

The Lean Startup seems to be heavily focused on adopting agile and lean principles in business development more than software or physical product development. In this research it would best serve if subject of the research would be trying to find new business opportunities rather than trying to find ways to improve how continuous product development could be improved in established operations environment.

2.6 Spotify model

During past couple of years, Spotify Model's popularity has increased significantly (VersionOne 2018). Although it is not a distinct framework, but more a rapidly evolving fusion of several agile and modern management practices.

Originally Spotify followed Scrum methodology until the company started to grow rapidly and some standard Scrum practices begun to get their way. They decided that Agile principles should have greater importance than obeying certain methodology or practices. This led to a course of development which Kniberg (2012) calls more still ongoing evolution than a big re-make of operating model. (Kniberg 2014.)

Spotify use quite common Agile practices like small cross-functional teams, called squads, which are self-organizing. Leaders communicate company level vision and what are the problems to be solved. Squads collaborate with each other to find the best possible solution, how to accomplish it and how to work together, leading into high autonomy. However, squads have mission which is aligned with product strategy and company short term goals which creates high alignment between loosely coupled squads. Organisation model is probably one of the most distinctive attribute to Spotify model. (Kniberg 2014.)

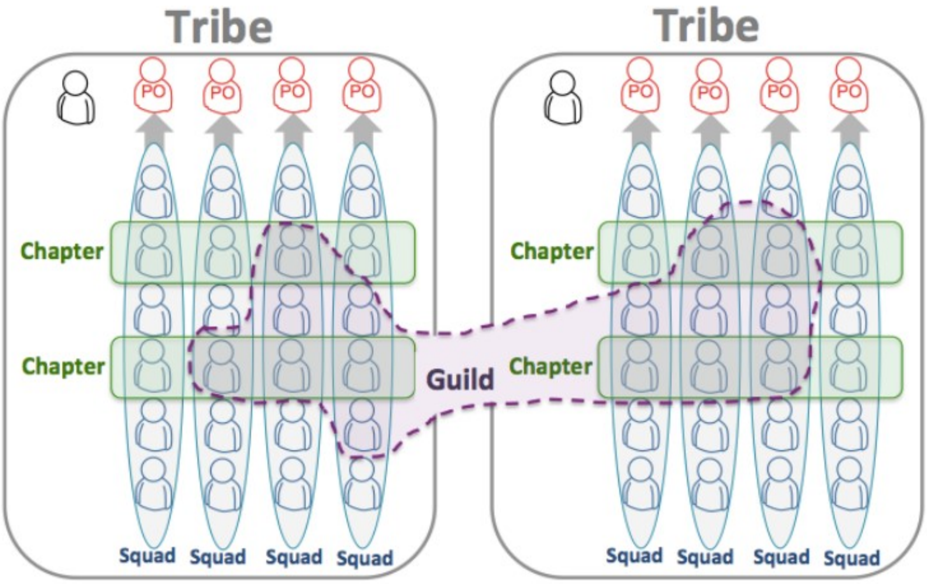


Figure 14. Spotify organization model (Kniberg & Ivarsson, 2012).

Squads are cross-functional feature teams with ownership of certain part of the solution responsible of full software lifecycle from design to release and even operation and maintenance related to feature. Every feature has a Product owner and team usually consist of frontend and backend developers, test and release specialists and an Agile coach which is analogous to Scrum Master to help the team to follow their chosen methodology and practices. Chapters are teams specialized in certain knowledge, like backend development. Chapter leads are line managers to people working with same or similar technologies, but in different Squads. Where Product Owners are responsible to decide which things are developed, Chapter Leads are responsible of coaching teams on how they should approach the request and actions resulting in equally high-quality code in every Squad. Tribe is a collection of features which usually form a natural larger entity like music player or recommendation engine. Finally, Guilds are voluntary communities of collective interest showcasing resolutions or mistakes and sharing knowledge on certain areas like security, Agile practices or development tools which usually span over multiple chapters. (Kniberg & Ivarsson, 2012.)

Generated code is always peer reviewed and all the code is available to everyone for alteration proposals. Spotify also makes releasing software as fast and automatic as possible with fit-for-purpose tools. This enables frequent and small releases. New code is rolled out gradually, first in a small subset of the whole environment serving users, simultaneously running old and new versions in parallel. Functionality and end user behaviour can be evaluated in real time to make decision whether to keep new version or return to the old version. These practices are familiar from Extreme Programming. (Kniberg, 2014.)

Generally, in Product Development Spotify follows Lean Startup methodology first defining a narrative and prototype how new feature or fix that should work. After that a Minimum Lovable Product, a derivative from Minimum Viable Product, is build and released. Based on careful analysis of predefined measure a decision to tweak or ditch the feature is made. Failing fast is seen as a possibility to learn fast which ultimately leads into fast improvement. Even Spotify CEO, Daniel Ek has made a statement that “We aim to make mistakes faster than anyone else”. (Kniberg, 2014.)

Spotify also tries to build an innovative culture by embracing experimentation, having 10 % of hack-time what people can use to develop whatever they like. And running hack weeks twice a year combined with demos and afterparty. These have resulted as several features used now in Spotify's music service. (Kniberg, 2014.)

Even though Spotify has been successful, very likely at least partially because of their working methodology, they advise other organisations not to copy what they have done but concentrate on experimenting with the correct fit between practices, organisation culture and current situation. As Spotify keeps growing, market and consumer habits changing they also feel a constant need to improve. Meaning change the ways things are done today to accommodate better fit-for-purpose ways to do things tomorrow. (Florian, 2016.)

2.7 Extreme Manufacturing

Even though Extreme manufacturing (XM) doesn’t top popularity lists or as such isn’t a distinct methodology but an assortment of Agile practices, it is highly interesting as it’s probably a most thoughtful and advanced approach to combine Agile Software Development methodologies and practices in building tangible, hardware objects. Joe Justice, a founder and CEO of Wikispeed and a major contributor of Extreme Manufacturing since 2006, is today also a member of Scrum Inc. developing methodology now called Scrum in Hardware further (Wikispeed n.d. & Scrum Inc. n.d.). Extreme Manufacturing’s focus on building cars is also rather fascinating from this thesis’s point of view.

Extreme Manufacturing base on 7 principles (Justice 2011):

- minimize cost to make changes to innovate quickly
- loose coupling enables making changes in parallel
- working collaboratively n shared space removes blocks quickly
- doing automated tests first quickly confirms improvement
- test are success criteria
- team morale is multiplier for velocity
- Iterations and stubs make for constant successes

Cost of change like changes in team, tooling materials, components even goals are seen as an impediment of innovation. If sunk cost is not material, it’s not prohibitive to change and makes use of new innovative approaches possible. As an example, Justice refers to a 2000-dollar generic CNC machine compared to a million-dollar equal proprietary solution used by automotive industry (Brown & Justice 2018) and learning how to use composites to build a car body, instead of subcontracting one. (Justice 2011.)

Loose coupling of components has been adopted from Object Oriented Architecture, where modularity is achieved through contract-first design between modules with known stable interfaces (KSI). As an example, a joint of suspension to chassis is standardized. This is also only point where proactive design for future requirements is done to avoid costly changes to interfaces (Brown & Justice 2018). Otherwise components are designed minimally to fulfil predefined tests. Additionally, existing designs, materials and components are re-used throughout the car to minimize costs. (Justice 2011.)

Extreme Manufacturing follows Scrum practices to have, if possible, team physically located at the same location at the same time. This enables pairing and swarming practices inherited from XP. Test Driven development is also a practice from XP, now limiting design efforts to a level fulfilling predefined acceptance tests. (Justice 2011.)

High team morale is achieved through tangible, well-defined short-term objectives and feeling constant progress iterative approach creates. If team morale is high, results are multiplied compared to low morale. Iterative approach and stubs, which are short-term workarounds for more permanent solutions. Like having blocks of tree instead of real suspension system to be able to demonstrate a to-be finished product in real life. (Justice 2011.)

All-in-all Extreme Manufacturing combines 12 co-existing compatible practices from Scrum, Extreme Programming and Object-Oriented Architecture as illustrated in Figure 15.

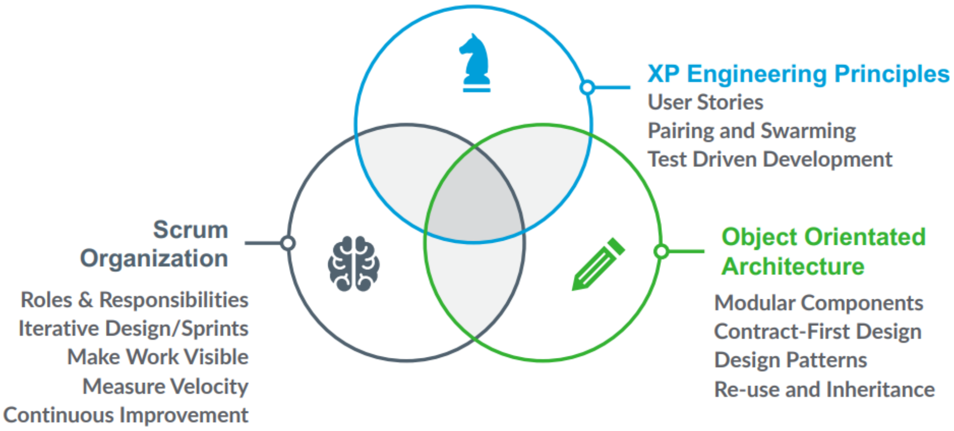


Figure 15. Extreme Manufacturing principles (Schwartz, 2017).

Additionally, Wikispeed uses Lean practices like 5S (Liker 2010, 150–151) to arrange working environment and have modular inventory. Whenever tools are located next to their consumables. Similar are located in same place, as few categories as possible. Everything is visually accessible like having transparent bins over drawers so that people can see the content. (Justice 2011.)

3 RESEARCH METHOD

3.1 Research problem and questions

The research problem was to understand could Agile methodologies provide benefits in Product Development Process of Discrete Manufacturing and Assembly Company.

The research questions were:

- is there a need to change the DMA Product Development Process
- could Agile software development methodologies be applied
- which Agile development framework fits best to DMA environment

– what kind of results (improvement) new operating model provides

Based on research and evaluation of suitable frameworks against research subject’s requirements there was a recommendation of actions, project to implement development process changes and measurement to evaluate benefits from the change.

Goals for process development were optimizing existing process, improving prioritization, increasing quality, elevating transparency and communication. If these goals are achieved, it will eventually have positive impact on employee satisfaction.

Success of this research was measured by positive trend between before and after change surveys related to key goals and keeping up or exceeding productivity by comparing experimentation period design release volumes to previous results.

3.2 Research methodologies

Lean management and Agile development falling under Systems Theory are abstract concepts even though they many times have physical embodiments. Those exist in consequence of human social activities and interactions. According to Karl Popper these kinds of phenomenon belong into World Three of objective knowledge having intersubjective ontology. (Anttila 2015, 41–45; Niiniluoto 1984, 319–321.)

As this research aimed to influence and develop organizational behaviour and researcher was actively contributing to change, research strategy quite naturally was Action Research where interests are critical and emancipatory with practical orientation. Unlike most research strategies, Action research prescribes researcher to influence on research subject. Research is tightly integrated into daily operation and all the participants in the process are subjects to research. Approach is empirical in a sense that it includes perception of research subject. It also has interpretative, hermeneutic attributes as researcher is required to interpret situation and factors influencing into it. (Anttila 2015, 439–443; Kananen 2014, 27–30.)

Action Research is a form of Blended Research allowing both qualitative and quantitative approach (Kananen 2014, 20–21). Qualitative semi-structured interviews were used to understand current situation, requirement for change and possible impediments for the change in DMA. This information combined with studies related to theoretical framework was used to evaluate optimal approach to reach goals. Quantitative Survey was used to evaluate project results.

During the research, abductive reasoning was used. Having an initial thought of issues and plausible approach to enhance current process was verified by having constant conversation with individuals and with groups of people in workshops (Anttila 2015, 118–121). Concurrently this information was used to select and evaluate different opportunities as future operating model for initial Plan-Do-Study-Act -cycle with primary goal of establishing continuously improving structured development process.

3.3 Data collection and analysis

Even though there was a quite good preconception about current situation of Product Development Process in DMA and several plausible solutions how to remedy issues caused by increasing demand, growing geographical spread and siloed teams combined with short time interval to adapt operating structures, personal semi-structured interviews were used to verify these preconceptions and understand root causes in more detail. Semi-structured interviews provide rich, descriptive information and doesn’t limit interviewee to a certain set of answers. Personal interviews were used to avoid influence of team dynamics and to create safe environment to express opinions. To avoid interviewer bias, open-ended questions were used throughout the interviews. (Anttila 2015, 195–201; Kananen 2014, 87–92.)

As qualitative research sample selection doesn’t exactly conform to theory of sampling, used sampling resembles at best discretionary sampling. Interviewees were selected so that every team, different nationalities and both specialists and managers contributing into Product Development Process were represented as they were available for the interview. Total number of 14 persons were interviewed to ensure high level of representation, even though answers started showing saturation already after 8–10 interviews. (Kananen 2015, 93–94.)

Interviews were held mostly face-to-face in DMA premises or some as a telephone interview. All interviews were recorded. Discussions about the key topics started from generic level gradually going into more detail simultaneously reflecting interviewee answers to make not-predefined additional questions. Interview language was either Finnish or English, depending on native language of interviewee. Recordings were initially transcribed by using speech recognition machine learning service (Google), but as the results didn’t have high enough accuracy, interviews were re-transcribed manually by using initial machine transcription. Intelligent verbatim transcription was used to have easy-to-read documents. Transcriptions were imported into cloud-based qualitative data analysis and research service (Atlas.ti) where it was coded based on themes rising from interviews. Coding of material was done in two passes to makes sure that themes occurring in later interviews were coded also in first interviews and to ensure coding consistency. Between the two passes some codes were split to have more detail analysis or merged if initial codes were near-similar.

Instead of trying to forcefully combine resulting 30 codes to higher level codes, data was analysed based on level 1 coding with spreadsheet application (Microsoft Excel) to bring up most discussed topics and concurrent topics occurring in same sentences. Based on this information a concept map of most pressing issues and likely resolutions was drawn. To confirm manual analysis, a cloud based semantic analysis tool (Infranodus) were used to analyse code data and build a full map of topics and their co-occurrence.

As there were no existing measurements which could have been used to verify effect of changes, a small survey was held before and after change. This approach is seen more reliable than conducting only one survey after the changes. Both surveys introduced five statements and each participant were requested to indicate their level of agreement with 5 level Likert scale from strongly disagree to strongly agree. Participants were also given an opportunity to leave question unanswered. There was also possibility to give generic open feedback. Latter survey also included a direct question whether participants felt that changes had improved the Product Development Process. Each survey questions were selected to depict one of the change goals, and questions can be found in Appendix 9.

A web-based structured survey was selected as a method, because it is easy to implement and provides numerical responses which makes relative comparison of before and after status straightforward. A web-based survey platform (Webropol) was used to conduct a survey for all identified 58 persons contributing to Product Development Process. A high number of survey participants was chosen to ensure reasonable amount of answers and chosen method didn't require additional work for increased coverage. Invitation e-mails (Appendix 4.) were sent late in the evening, so that participants would see those on top of their mail list in the morning. Participants were given a week to answer to survey. They were also notified three days before and again on the morning of the day the survey closed in the midnight. First survey was held during the week before and second survey was done during the ninth week after the implementation of new method. Quite long time between the surveys was planned to give people time to orientate to new way of working and to be able to have one continuous improvement session to re-iterate process.

Analysis of the results was done by comparing arithmetic means and variance of before and answer pairs of each question separately. As suggested by Anttila (2015, 246–250) Student's T-test were used to verify whether survey result pairs had statistically significant difference. A two-sample T-test assuming unequal variances were used, even though both surveys received twelve answers, but it was likely that respondents were not the same in both surveys. Survey results were also visualized with line chart to help analysis.

3.4 Reliability and validity of research

Credibility of the qualitative research was sought with sufficiently large number of interviewees leading into repetition in responds and comprehensive representation of different parties related to process. This approach also ensured that single interviewee didn’t have too much weight in synthesis.

Due to the nature of qualitative research, ensuring its validity and reliability is challenging and even more so to Action research which aims to change of research subject (Kananen 2015, 125–126). Mäkelä (1990, 48) has suggested having following criteria’s when evaluating qualitative research:

- sufficiency of material
- coverage of analysis
- evaluation and repeatability of the analysis

According to Kananen (2015, 134–137) focus should be put on:

- detailed documentation of results, methods and data collection
- confirmation of the research subjects for credibility
- detailed documentation of starting point for transferability
- use of material or methodological triangulation
- vindication of how interpretation has done and how it is extracted from material

The validity of the study was verified by evaluating different research-theoretical approaches to the problem and excluding the alternatives that were not relevant to the problem in question. Despite having a quite clear preconception of the issues and likely root-causes, a semi-structured interview was used to avoid possible bias of research subscriber. In addition, attention was paid to non-conducting interview practices, transcription and coding of material. The analysis was done using generally well-proven methods with verification from automated code analysis. Methodological triangulation was used as verification of results were analysed with survey, instead of having second round of interviews which provides additional confirmation of validity. Finally, research subscriber satisfaction reassures research validity.

The reliability of the research was assured by careful and accurate interview documentation and by striving for a consistent implementation both from content and interview environment point of view. Individual variation in responses was neutralized by the number of interviewees, so the results of the individual interview did not gain significant weight in the results. The interviews can be similarly replayed, but as the situation in the organization changes quickly, the results are only valid for a limited time. On the other hand, by conducting the survey again with same content for example every six months, it is possible to evaluate the direction and velocity of evolution. Replenishing focus areas requires interviews to be repeated or an enlightened assessment of the changed state.

3.5 Subject of research

Discrete Manufacturing and Assembly Company (DMA) was founded in early 2000's, but demand begin to surge couple of years ago. During past couple of year company has grown from virtually zero to a 150 person, nearly 100 m€ operation. Despite strong growth, DMA has been able to sustain 12–13% operating profit level, which can be considered a good achievement. Most of the diverse personnel consisting around 10 nationalities are Finnish and new assembly line is starting abroad

Even though DMA has been successful, they are always seeking opportunities to enhance their productivity, especially on product development which is in the end the key to sustainable success. Successful business has been possible so far because of quite small, highly motivated and skilled group of individuals with flexible working methods. To be able to build sustaining success more structure in working methods without losing agility is sought after. Also growing number of dispersed resources are requiring increasing communication, transparency and optimized process to be followed by all the contributors, which needs to be addressed.

4 EMPIRICAL STUDY

4.1 Survey of current state

A semi-structured interview was used to survey current state of the Product Development Process (PDP) in Discrete Manufacturing and Assembly Company. Before actual interviews there was a kick-off meeting at 27.9.2018 for all persons contributing to or being a stakeholder for PDP about process development initiative and related thesis work. All the participants received an invitation letter (Appendix 3) in their meeting reservation. The goal of the interviews was:

- to understand current state of product development process
- to verify need of change
- to understand possible impediments for the change

These were also the themes used in interviews which were broken down into questions drilling in to details of each topic. Additional, ad-hoc questions were also used based on individual discussions.

First round of six face-to-face interviews were conducted 2.10.2018 in DMA headquarters. Version 1.0 of interview questions (Appendix 2) were used. Interviews were held in Finnish or English, depending on interviewee native language. Interviews were recorded and transcribed during next two weeks with machine learning-based speech recognition service and manual correction, because machine transcriptions too low quality. During this process, recordings were listened several times. Transcriptions were sent to interviewee for possible corrections and remarks. After that all interviews were initially categorized based on discussion content.

Second round of six face-to-face interviews were conducted 15.10.2018 in DMA headquarters. Questions were subtly refined based on first round of interviews and version 1.1 of interview questions (Appendix 2) were used. Interviews, transcription and categorisation followed same procedure as in first round. Two additional phone interviews were arranged at October 26th and 30th 2018 making total number of interviews 14, despite there was distinctive saturation of interview content already during second round. Otherwise the process for these telephone interviews remained same.

Interview categorization was done twice on the material, a second pass merging some similar codes. This also increased code quality as couple of the categories were raised in later interviews, even though they existed already in the early ones, but were left unnoticed.

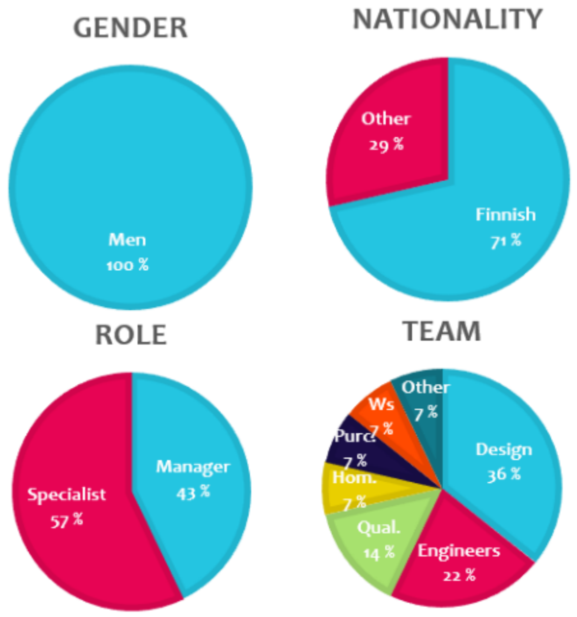


Figure 16. Interviewee demographics.

During the interviews, nearly 9 hours of conversations were recorded as interviews lasted from 25 to 55 minutes averaging 35 minutes. After transcription there were 76 pages with over 40’000 words material for categorisation. Categorisation resulted 668 quotes with total of 1446 tags from 30 categories. Most of the categories highlighted discussion topics, but “Change”, “Benefits” and “Impediments” were used in combination with topics to highlight areas in need of change, change benefits and change impediments.

After categorisation a word cloud (Figure 17) of category incidence was created to visualise most discussed topics. Many of the co-exist in same discussion or sentence having clear relation with each other. Most discussed topics were communication (9,0%), management (8,8%) and process (7,8%). In the end, everything starts from the management by its contribution to what needs to be accomplished (objectives, prioritisation), by whom (people, roles), by when (schedule, prioritisation), where (facilities), why (mission) and how (process, communication, tools).

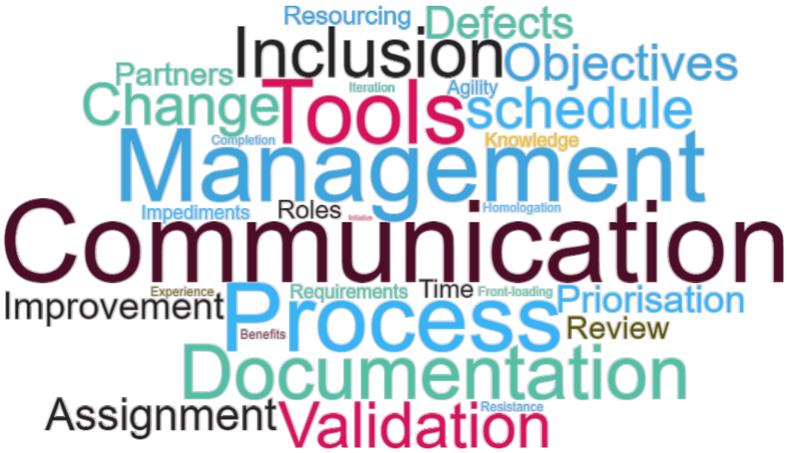
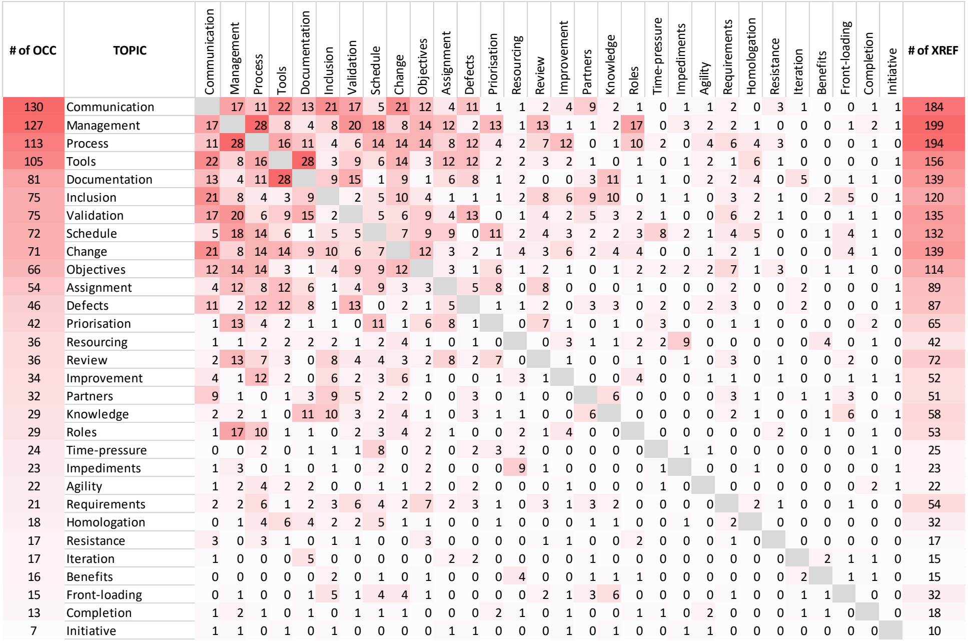


Figure 17. Interview category word cloud.

To have more understanding topic relationships a cross-reference heat map (Figure 18) was created visualizing topics occurring in same quotes. Unsurprisingly, management had the most cross references (8,5%), processes being second (8,3%) and communication third (7,8%).



# of OCC	TOPIC	Communication	Management	Process	Tools	Documentation	Inclusion	Validation	Schedule	Change	Objectives	Assignment	Defects	Priorisation	Resourcing	Review	Improvement	Partners	Knowledge	Roles	Time-pressure	Impediments	Agility	Requirements	Homologation	Resistance	Iteration	Benefits	Front-loading	Completion	Initiative	# of XREF
130	Communication		17	11	22	13	21	17	5	21	12	4	11	1	1	2	4	9	2	1	0	1	1	2	0	3	1	0	0	1	1	184
127	Management	17		28	8	4	8	20	18	8	14	12	2	13	1	13	1	1	2	17	0	3	2	2	1	0	0	0	1	2	1	199
113	Process	11	28		16	11	4	6	14	14	14	8	12	4	2	7	12	0	1	10	2	0	4	6	4	3	0	0	0	1	0	194
105	Tools	22	8	16		28	3	9	6	14	3	12	12	2	2	3	2	1	0	1	0	1	2	1	6	1	0	0	0	0	1	156
81	Documentation	13	4	11	28		9	15	1	9	1	6	8	1	2	0	0	3	11	1	1	0	2	2	4	0	5	0	1	1	0	139
75	Inclusion	21	8	4	3	9		2	5	10	4	1	1	1	2	8	6	9	10	0	1	1	0	3	2	1	0	2	5	0	1	120
75	Validation	17	20	6	9	15	2		5	6	9	4	13	0	1	4	2	5	3	2	1	0	0	6	2	1	0	0	1	1	0	135
72	Schedule	5	18	14	6	1	5	5		7	9	9	0	11	2	4	3	2	2	3	8	2	1	4	5	0	0	1	4	1	0	132
71	Change	21	8	14	14	9	10	6	7		12	3	2	1	4	3	6	2	4	4	0	0	1	2	1	0	0	0	4	1	0	139
66	Objectives	12	14	14	3	1	4	9	9	12		3	1	6	1	2	1	0	1	2	2	2	2	7	1	3	0	1	1	0	0	114
54	Assignment	4	12	8	12	6	1	4	9	3	3		5	8	0	8	0	0	0	1	0	0	0	2	0	0	2	0	0	0	1	89
46	Defects	11	2	12	12	8	1	13	0	2	1	5		1	1	2	0	3	3	0	2	0	2	3	0	0	2	0	0	0	1	87
42	Priorisation	1	13	4	2	1	1	0	11	1	6	8	1		0	7	1	0	1	0	3	0	0	1	1	0	0	0	0	2	0	65
36	Resourcing	1	1	2	2	2	2	1	2	4	1	0	1	0		0	3	1	1	2	2	9	0	0	0	0	0	4	0	1	0	42
36	Review	2	13	7	3	0	8	4	4	3	2	8	2	7	0		1	0	0	1	0	1	0	3	0	1	0	0	2	0	0	72
34	Improvement	4	1	12	2	0	6	2	3	6	1	0	0	1	3	1		0	0	4	0	0	1	1	0	1	0	0	1	1	1	52
32	Partners	9	1	0	1	3	9	5	2	2	0	0	3	0	1	0	0		6	0	0	0	0	3	1	0	1	1	3	0	0	51
29	Knowledge	2	2	1	0	11	10	3	2	4	1	0	3	1	1	0	0	6		0	0	0	0	2	1	0	0	1	6	0	1	58
29	Roles	1	17	10	1	1	0	2	3	4	2	1	0	0	2	1	4	0	0		0	0	0	0	0	2	0	1	0	1	0	53
24	Time-pressure	0	0	2	0	1	1	1	8	0	2	0	2	3	2	0	0	0	0	0		1	1	0	0	0	0	0	0	1	0	25
23	Impediments	1	3	0	1	0	1	0	2	0	2	0	0	0	9	1	0	0	0	0	1		0	0	1	0	0	0	0	0	1	23
22	Agility	1	2	4	2	2	0	0	1	1	2	0	2	0	0	0	1	0	0	0	1	0		0	0	0	0	0	0	2	1	22
21	Requirements	2	2	6	1	2	3	6	4	2	7	2	3	1	0	3	1	3	2	0	0	0	0		2	1	0	0	1	0	0	54
18	Homologation	0	1	4	6	4	2	2	5	1	1	0	0	1	0	0	0	1	1	0	0	1	0	2		0	0	0	0	0	0	32
17	Resistance	3	0	3	1	0	1	1	0	0	3	0	0	0	0	1	1	0	0	2	0	0	0	1	0		0	0	0	0	0	17
17	Iteration	1	0	0	0	5	0	0	0	0	0	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0		2	1	1	0	15
16	Benefits	0	0	0	0	0	2	0	1	0	1	0	0	0	4	0	0	1	1	1	0	0	0	0	0	0	2		1	1	0	15
15	Front-loading	0	1	0	0	1	5	1	4	4	1	0	0	0	0	2	1	3	6	0	0	0	0	1	0	0	1	1		0	0	32
13	Completion	1	2	1	0	1	0	1	1	1	0	0	0	2	1	0	1	0	0	1	1	0	2	0	0	0	1	1	0		0	18
7	Initiative	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0		10

Figure 18. Category incidence and cross-reference heatmap.

Feedback related to management, processes and communication was heavily intertwined. Typical comments related to management highlighted absence of clear responsibilities and objectives, missing measurement for operational improvement or at least communication of those have failed. As an example of comments:

- *I don't think it's well managed today. It's more based on each person's own though, how his work should be done.*
- *This has been a point of contention lately, because we don't have anyone to do validation.*
- *We have some safety-critical components that have been planned half a year ago. But we haven't been able to release them because no one has given permission.*
- *Before we had a Development Meeting deciding task prioritisation and deadlines. But we haven't had that for some time now.*
- *There is no agreed way. It's just based on discussion and persons skills and own conclusions.*
- *Objectives and measures are missing. We should be setting some goals and have follow-ups and reviews.*
- *We need coherent, managed actions. Not so much how it is today, that something is done, then a small group of people is doing all the tasks.*
- *There is no long-term plan for things that cannot be done now but could be done after four months to meet this need.*
- *Strategy should define what role it is, but we haven't really documented roles.*
- *I would still emphasize the fact that we need a person who manages the process from start to finish.*

Based on DMA's documentation, sub-processes like design, purchasing and manufacturing are documented on a reasonable level, but it seems that those are quite new and not well-known. Sub-processes are heavily siloed and there is no defined method how to manage end-to-end process. When combining interview feedback and documentation, it seems that interest for documentation comes more from compliance point of view than operational excellence and continuous improvement. Some examples of process related comments:

- *This current product development process is a bit confusing to me.*
- *The practice of who makes decision and when it is done is unclear. It's hard to get information when some part is changed from test to release. The way the information is flowing doesn't work well.*
- *There is a lack of clear structure how we work. There will always be deviations, that's normal. But currently half of the tasks have process deviations and half is done according normal operation.*
- *You cannot say that there is a certain process to be followed at all times. Approach varies based on situation.*
- *In my opinion, it's coming more from the spinal cord. There are certain steps where the process is being followed, but then there are steps where they might be followed or not.*
- *And how to get feedback when there is not even any tool for that. Information is scattered, and it doesn't help that we don't have the process chain in one place.*
- *But I haven't noticed anyone going it [development list] through during past three months.*
- *So, the idea of how we should work is there, but today's practical implementation does not go that way. It is not even near yet.*
- *Process doesn't probably have a real owner. That's quite likely a bad side in it.*
- *At the moment there's no real structure to how anyone picks up a job and then does the job and then finishes the job.*

Most of the people felt that communication within same team was functioning well even though it was quite heavily face-to-face, and e-mail focused, which will not leave a publicly available document trail like team discussion groups. Sometimes these e-mail or face-to-face discussions are in Finnish, which makes it hard to join to discussion for non-Finnish speaking team members. Then again, information flow between different teams was seen problematic, usually requiring excessive activities to have. It starts from begin of the process, about priorities in high level and each task, having all necessary information in assignments, exchanging knowledge between teams and suppliers during design-phase, having delivery time information for parts, having information about what have changed in the part to quality control and testing and finally getting feedback from tests to design for future improvement. Comments varied as follows:

- *The most important method is talking face-to-face, as everybody is very close to each other. E-mail comes second.*
- *I would say just usually email, but people always get left out by emails. And you never get entire picture what's happening.*
- *Most of the communication is informal, based on personal contacts and conversations. There's no continuous meeting practice.*
- *Current operating model produces such feedback that people cannot plan their own work.*
- *A lot of time it just takes forever for me to gather all the information I need before I start doing something.*
- *I feel like the only way to talk with suppliers is through our purchasing department.*
- *As soon as you finish the component, it goes to purchasing, and that's the last thing you hear about it.*
- *Operations Director does a Monday morning meeting, like at 10 on the Monday everyone goes over to the meeting room. He will come in and will just let us know what's happening in near future.*
- *I would like to see that maybe not as a meeting, but it'd be nice to have a running list or something you can easily check on see on what development stage that part is in.*
- *Communication is the biggest issue currently. Information is moving only between those persons who work on a certain task.*

Most of the constructive feedback was related to not having adequate operational structures and practices in place considering size of the operations. Many of the other topics are related to these three primary themes. Tools should support the end-to-end process, provide measurement against objectives, help people communicate and find information. Objectives, prioritization and reviews are forms of management practices. Roles, scheduling, validation and improvement are process related practices. Inclusion, engaging early (front-loading), knowledge and documentation are heavily communication related topics.

Quite clearly, the huge time-pressure to be able to release competitive product combined with rapid growth of organization to demand and competition were prioritized over structures required to run larger operation. Achieved success has been based on visionary leadership and highly skilled and motivated individuals with a common mission to overcome obstacles. But it is a good question whether it is sustainable and scalable in the long term without having required structures in place.

Even though the purpose of these interviews where to identify areas in need of change and as such the responses seem to be negative, there was a constant feeling of pride related to interviewee contribution and achieved results as a team. Every interviewee commented like this to a question about strengths of current operating model:

- *Agility and absence of bureaucracy. We are able to make big changes rapidly.*
- *Our ability to react rapidly. We are really flexible and fast when required. It doesn't take long to get new parts from design to test.*
- *When needed, our ability to react is astonishing.*
- *We can react quickly to high priority things. If we need to fix something for the next shipment, we drop everything else and just concentrate on that.*
- *Flexibility of people and operating model is surely our strength. We can make changes very rapidly.*
- *We are flexible and rapid when required.*
- *We are able to make changes rapidly.*
- *Agility and initiative are great our traits. And we have loads of know-how.*
- *We are able to withdraw and rethink wrong decisions.*

To understand the obvious need for change in more detail, ten most occurring change discussion topics were chosen as a starting point. Out of the remaining topics, all which had more than 5 cross-references to any of the Top 10 were additionally selected to increase context. Front-loading was added to list later-on as its importance for high quality designs seemed obvious.

Top 10 Topics	Top 10 related	Other
Communication	Assignment	Front-loading
Process	Defects	
Tools	Knowledge	
Objectives	Partners	
Inclusion	Prioritisation	
Documentation	Review	
Management	Roles	
Schedule		
Validation		
Improvement		

Table 3. Key change topics based on interviews.

These key topics of issues were then put into a concept map to visualize cause and effect flow. Topics were positioned manually closer to other most cross-referenced topics with arrows indication cause and effect direction.

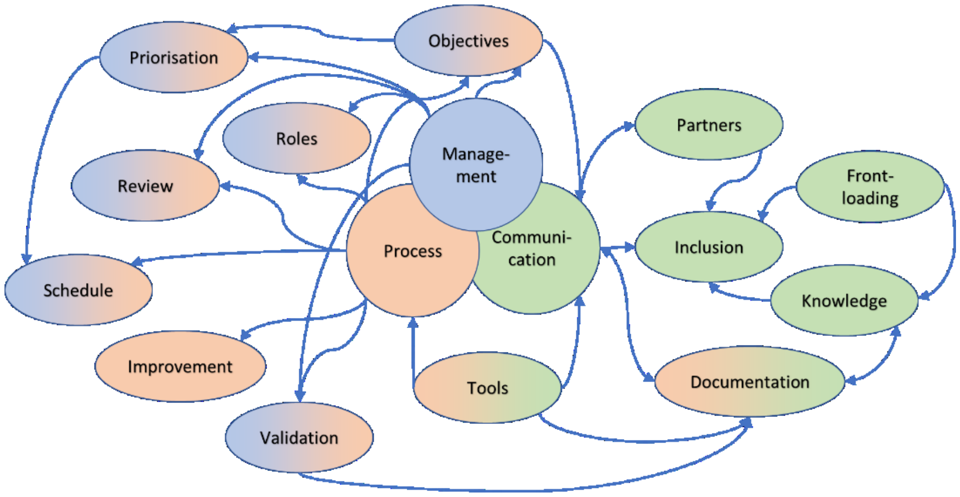


Figure 19. Concept map of the categories requiring the most change.

Based on concept map (Figure 19) management contributes into definition of processes and ensures high level communication in the company. In this case contribute doesn't necessarily mean that management does those things, but more likely delegate some of the activities to persons or teams. Processes and communication policies bind also management and that's why decision is required to confirm the policies. Management defines and communicates clear objectives. Process definition should include roles, review policies, scheduling, validation and most importantly method of continuous improvement. Management prioritises assignments according to objectives and contributes in review and validation to provide support for organisation and to promote healthy governance culture. Testing and validation should transform into documentation which should be communicated to everyone. In turn communication is a form of documentation. Tools should support following processes, communication between stakeholders and to creation, finding and updating documentation. Communication throughout the process should be inclusive to all members contributing to process, including suppliers making sure that full potential of cumulative knowledge is used. This combined knowledge should be documented to help in induction and future redesign activities.

This interpretation follows quite closely Leavitt's Diamond (Leavitt 1972, 262–265) proposing all organisational systems consisting of four elements: People, Task, Structure and Technology acting in certain environment. In this framework People represent employees, their skills, competencies and knowledge. Tasks include definition of how things are done and what are the goals organisation is trying to achieve. Structure defines how organisation is managed, coordinated and organised. And how communication is done between different contributors. Technology is a component, which supports all aforementioned elements.

Lastly to verify conclusions done based on interview data, per quote coding of interviews were imported into cloud based semantic analysis tool (Infranodus) which uses Louvain community detection algorithm (Blondel, Guillaume, Lambiotte & Lefebvre 2008) to combine topics based on their co-occurrence. This automated analysis (Figure 20) supports mostly the results of manual analysis, despite some minor variance most likely caused by system using full data set whereas manual analysis was intentionally done with suppressed amount of data by limiting analysis only to highest number of occurrences and cross references to make analysis more manageable.

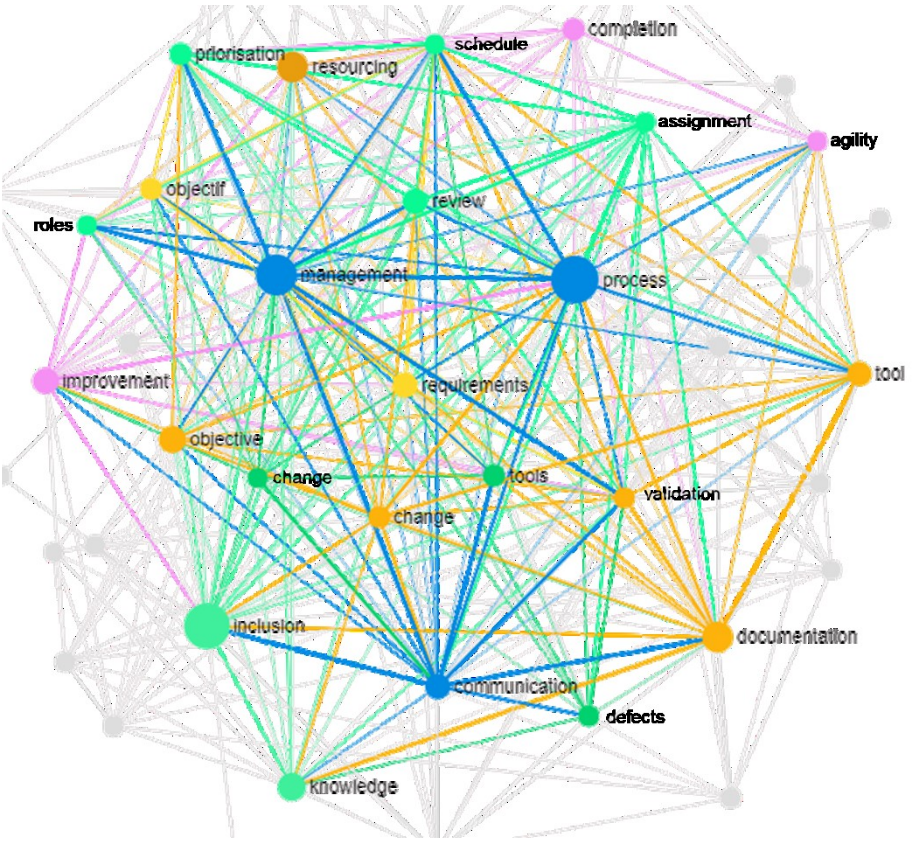


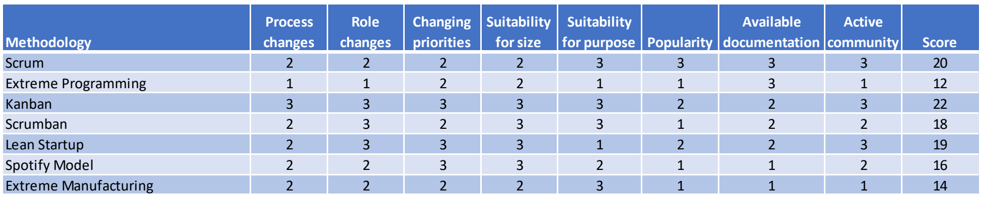
Figure 20. Automatic semantic analysis of interview quotes coding.

To summarize change focus:

- Redefine and implement end-to-end product development process
- Define and communicate objectives and follow results
- Build-in natural cross team communication into processes
- Provide tools to support process, communication and knowledge

4.2 Choosing a methodology

The goal of choosing the proposed methodology to DMA was to find a most suitable methodology which would not require extensive role and immediate process changes which may interfere current progress or in worst case stall development. The methodology should fit for the organization size and purpose and support constantly changing priorities. Additionally, it should be popular enough to have external support, comprehensive available documentation and active community developing methodology.



Methodology	Process changes	Role changes	Changing priorities	Suitability for size	Suitability for purpose	Popularity	Available documentation	Active community	Score
Scrum	2	2	2	2	3	3	3	3	20
Extreme Programming	1	1	2	2	1	1	3	1	12
Kanban	3	3	3	3	3	2	2	3	22
Scrumban	2	3	2	3	3	1	2	2	18
Lean Startup	2	3	3	3	1	2	2	3	19
Spotify Model	2	2	3	3	2	1	1	2	16
Extreme Manufacturing	2	2	2	2	3	1	1	1	14

Table 4. Evaluation of different methodologies against requirements.

Evaluation is subjective researcher’s view based on experience and material read during this research. This evaluation is heavily context specific and thus doesn’t generally suggest any preference between methodologies. It is also good to remember that list of methodologies is not exhaustive.

Scrum is a very popular methodology, with great documentation available and vast, active community exchanging information. There are many examples of using Scrum outside software development, even in personal life of people, but before mastering the method it emphasized following as strictly as possible prescribed ceremonies, roles and artefacts. Requirement for cross-functional team(s) doing end-to-end the process in question is rather challenging. Scrum would also need some scaling method to support up to 30–50 participants. Finally, protection of sprints and development scope is something which works against set requirements.

Extreme Programming’s popularity has been decreasing for a decade and many of its key practices has been adopted by more popular rivals. At the same time community activity has been declining, but documentation is on the benchmark level. XP has quite prescriptive process and roles, it protects iterations from changes and is heavily software development focused. Despite of that, many of its practices are something which should be experimented by DMA.

Kanban isn’t quite popular as Scrum, but it’s steadily growing its share and at the same time two thirds of VersionOne’s survey respondents use the Kanban board. It is straightforward to implement, as it doesn’t require immediate changes how organization operates, and it is possible to reprioritize requests even within the process by expediting. Even though many agile methodologies prescribe small teams, Kanban can easily grow by having specialized team throughout different process phases. Available documentation could be better, but active community makes finding help effortless.

Scrumban, as a fusion of Scrum and Kanban, or a way to transit from Scrum to Kanban seems a somewhat strange option if you haven't use Scrum previously. It emerges into current operating model a little easier than Scrum, but its popularity is low which also visible in available documentation and community.

Lean Startup is interesting newcomer, but it is focused on higher level to business development and transformation in general. Available documentation doesn't match the benchmark, but active community surely patches that shortage.

Spotify model is quite proprietary and as Spotify's representatives declares, no one should try to copy it, but rather try to find ideas to experiment in own organizations. Its popularity is low and available documentation restricted and incomplete focusing on certain parts. Despite of that there seems to be quite active and loud group of followers for this methodology.

Extreme Manufacturing or Scrum for Hardware is in many ways intriguing concept, although it shares same deficiencies with Scrum. Documentation is still quite incomplete, and community is rather small. But I wouldn't be surprised if this gets a lot of traction in manufacturing industry during following years.

Based on this evaluation and as objective discussion as possible with DMA key representatives related to product development process, Kanban was unanimously chosen as a methodology to be experimented. Experimentation scope was whole Product Development Process from idea generation to the point where new part is review and accepted to be used in the product.

4.3 Implementation plan

As Andersson (2010, 175–176) has generic predesigned action point list for starting Kanban in organisation, it was used a little modified as an implementation plan:

- Agree on a set of goals for introducing Kanban
- Map the value stream
- Define points where you want to control input
- Define an exit point beyond which you do not intend to control
- Define a set of work item types
- Define classes of service
- Create a board template
- Meet with the stakeholders about policies and coordination
- Finalize and create electronic board
- Educate the team on the new board

Implementation started with workshop immediately after decision of further progress was made to gather required information to ignite planning. Seven first items were drafted in collaboration of e-mails and virtual meetings with key stakeholders.

4.3.1 Agree on a set of goals

Considering the findings and focus areas in chapter 4.1 and generic goals for use of Kanban system (Anderson 2010, 169–174), following set of five DMA specific goals were set.

We need to **optimize process** to ensure short lead time, increase predictability and avoid unnecessary work (waste). This will be done by use of existing process, but involving everyone by expecting justified, commonly approved improvement proposals which experimentation is embraced. Results will be measured to empirically validate impact of change. Workflow will be visualized to identify bottlenecks and slack (idling). Limiting work in progress will be used to decrease lead time.

We want to **provide transparency** to build trust between team and stakeholders, set right level of expectation, get feedback to learn and help induce new team members. This will be achieved by making workflow and all assignments and those progress visible. We need to document activities in all the process phases and constantly communicate results and achievements to all process contributors and participants.

We need to **increase quality** to ensure effectiveness, increase predictability and avoid rework (waste). This will be pursued by identifying and documenting defects, issues and blocks and remediating root causes for these. Limiting work in progress will increase focus for smaller number of defects. Clear validation criteria’s and communication of test results will boost learning. Definition of done’s and predefined quality criterions must control flow within the process. Task lists are recommended to off-load memory for value adding purposes.

We must **simplify prioritization** to ensure maximum value, minimize risk and cost, and remove waste generated from constant reprioritization. This will be done by prioritization based on common objectives and cost of-delay, in which the cost is an equivalent to lost seconds or positions. All the assignments will be processed based on simple, predefined rules.

Ultimately, we are committed to **improve employee satisfaction** to reduce retention and thus losing knowledge, increase workplace attractiveness and sustain high productivity over long period of time. These will be achieved by improving work-life balance underpinned by reliability, providing feedback and showing respect for everyone’s contribution. And everyone should remember that DMA is building a winning team, not just individuals.

4.3.2 Map a value stream

Based on discussions during interviews and in Kanban implementation kick-off, and existing documentation a rough illustration (Figure 21) of the product development process was documented.

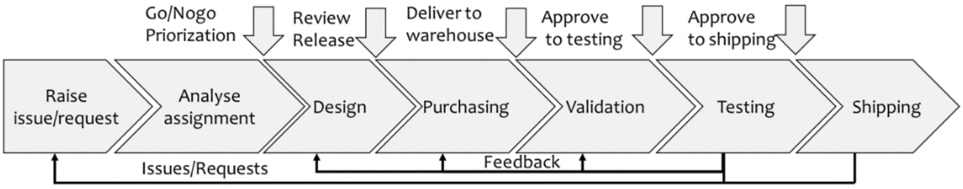


Figure 21. DMA Product development process value stream map.

Process start from raising an issue noticed in tests or by customers, or a request related to development ideas from process stakeholders or designers. After documenting a request an assignment analysis is performed for Development meeting Go/No-go decision making and prioritisation. Although, there is a common view that some of the requests are going directly to design, if those are quite small and cost of requested change is marginal. Also, Development meeting isn’t well established so many of the requests just go to Technical Director or Design Team Lead for approval.

After approval, request goes into design engineer who designs the part, sometimes by himself and sometimes using colleagues to evaluate the finished design. Release review is done by Technical Director or Design Team Lead giving approval to order the newly designed part.

Released parts designs go next to Purchasing, which is sometimes contacted already during design phase, for supplier selection and possibly requesting supplier view of designed part. Purchasing then sends a purchase order and design drawings and other technical material to supplier. Between purchasing and validation phases parts are manufactured by supplier, which could take something from couple of days to several weeks depending on manufacturing method used.

Most parts arrive at warehouse, they are quality checked by quality engineers and sent to workshop for test installation and after that to planned test location. In urgent situations, parts are sent directly to testing site for installation and testing.

As parts arrive and are checked, a test plan for a part is done and tests are conducted for set of parts during the test-days by test engineers. If parts do not fulfil the test qualifications, redesign request is sent to Design Team. If part do not fulfil the quality criteria's or specifications, information is sent back to purchasing and validation.

After successful tests, a Technical Director decides whether new part is approved to be used in the shipped products and setup of the product is changed accordingly. This activity finishes the Product Development process.

Two activities, a part homologation, meaning requesting acceptance to use from authorities, and production planning is done throughout the process phases from design to product-ship approval.

Natural entry point for the Product development process was in the assignment analysis phase and exit point in accepting the part to be used in the shipped products. Of course, only design phase could have been one option for optimisation scope, but as the whole end-to-end process is quite highly intertwined it made more sense to treat it as entity.

4.3.3 Define work item types and classes of service

Work item type are left intentionally to only three:

- Feature request
- Change request
- Issue

in the initial experimentation to gain simplicity.

A feature request is large work requiring in the minimum a working day to design a requested part. A change request is small part taking only hours to design. Assumption is that the lead time, elapsed time from part being moved from backlog to final done, is significantly longer for feature than change requests.

An issue is raised every time there is an impediment preventing fulfilling a request, called block. Or if there is a need to return to any previous process phase because of the quality issue or part not being able to qualify in tests.

Main reasons for this approach was, that there was a will to use work item type information for more accurate estimation of lead time in the future. But having not enough information today to be able to categorize the request didn't support this practice. It was also a fear that making a request becomes too complicated, which would eventually lead poor information. This way we also avoided need for analysis of current demand for different kind of work item types which could have proved hard as there was quite limited information for the analysis. Allocation was done by round robin fashion, so that every other request should be change request.

In the future, it will be good to experiment whether for example defected parts should have their own work item type. Or different generations of the product being designed. Usually work item types are used to allocate or guarantee certain number of development resources to certain types of work. In software development, for example, a certain number of persons or working time is usually allocated to fixing bugs so that number of bugs in the software is gradually decreasing.

Based on discussions, three types of priority classes were chosen:

- Expedite
- Fixed Date Delivery
- Standard

Expedite is an urgent priority class request, which needs to be fulfilled as rapidly as possible. When an expedite request arrives to to-do, or any buffer state, it will be immediately processed by the team in question. A request which get superseded is marked blocked with issue description linked to expedite. Also expedite may override any work-in-progress limits. Usually these requests are notified by supervisors to suitably skilled persons for the task. All team members are also informed in daily standup meetings. There should be as few as possible expedites in the system simultaneously as they disturb the flow.

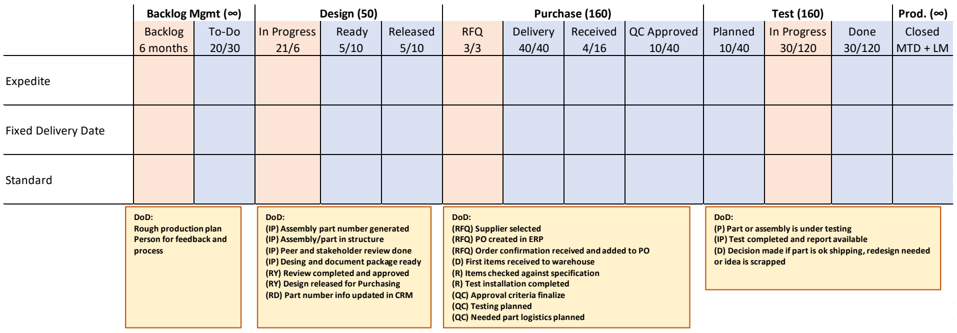
Fixed Date Delivery is a scheduled class request, meaning that benefits will realize only in specific event or date. A request with shortest time left to delivery date, will be prioritized over other requests, except expedites. Rationale is that new parts should not be designed to wait in warehouse for their use as storages are waste, but those should be designed and manufactured as late as possible. But to make sure that new parts are finished according scheduled delivery, those will have higher priority. That is why fixed date delivery requests should have lead time estimation and they require additional analysis when they are opened into backlog. If needed a fixed date delivery request may be escalated to expedite priority.

A standard class requests are served last so that oldest request in the system will be always served first. In this way lead time deviations are kept in the minimum. If needed a standard request may be escalated to expedite priority.

4.3.4 Create a board template

Based on current process described in Chapter 4.3.2, a Kanban board template started to formulate. It was decided already in the beginning that because of the team contributing to process was scattered around Europe and some of the participants might join from test locations, experimentation will be done with electronic board, which could be easily integrated in collaboration tools teams were already using.

A finished board was a result of three separate workshop sessions, last one including a process dry run, with couple of different types of assignments done in past couple of months. Based on feedback from these sessions with participants from development management team and PDP process contributors, it was decided to omit several active work states so that request is moved one buffer state to other and work is done in between. These work phases are review or validation type of tasks which take around 15–30 minutes to fulfil. Reason for this was to avoid unnecessary double-hopping of phases and make board smaller and thus easier to fit in TV and computer screens.



	Backlog Mgmt (∞)		Design (50)			Purchase (160)				Test (160)			Prod. (∞)
	Backlog	To-Do	In Progress	Ready	Released	RFQ	Delivery	Received	QC Approved	Planned	In Progress	Done	Closed
	6 months	20/30	21/6	5/10	5/10	3/3	40/40	4/16	10/40	10/40	30/120	30/120	MTD + LM
Expedite													
Fixed Delivery Date													
Standard													
	DoD: Rough production plan Person for feedback and process		DoD: (IP) Assembly part number generated (IP) Assembly/part in structure (IP) Peer and stakeholder review done (IP) Desing and document package ready (RY) Review completed and approved (RY) Design released for Purchasing (RD) Part number info updated in CRM			DoD: (RFQ) Supplier selected (RFQ) PO created in ERP (RFQ) Order confirmation received and added to PO (D) First items received to warehouse (R) Items checked against specification (R) Test installation completed (QC) Approval criteria finalize (QC) Testing planned (QC) Needed part logistics planned				DoD: (P) Part or assembly is under testing (IP) Test completed and report available (D) Decision made if part is ok shipping, redesign needed or idea is scrapped			

Figure 22. DMA Kanban Board Template (Appendix 7).

Product Development Process was split into 5 major and 13 minor phases. Each minor phase has work in progress (WIP) limit and definition of done (DoD). WIP limits were left intentionally quite loose, but at the same time giving a prominent statement to pursue significantly smaller batches and shorter lead time. Main principal was to have as few rules as possible. Rules should be added later by teams themselves based on experimentation, not only by directive of authority.

Process begins with backlog phase, where everyone is encouraged to raise new ideas. Requests have predefined format and they need to contain relevant information required to estimate benefits and start design work. As such, active backlog grooming should be avoided, but people raising requests should promote their ideas to Development meeting members, who makes priority decisions. Fixed date delivery requests need to be analysed by experienced engineer to evaluate plausible lead time. Development meeting will decide which requests will provide greatest relative benefits e.g. benefit per cost and elevate such requests from backlog to to-do. Items which will be elevated must have rough production plan done including target test, target product shipment, test part quantities and preliminary approval criteria defined. Additionally, persons must be assigned for feedback from purchasing, quality, workshop and engineering so that they know to expect the request and can contribute and support early in the process. To keep the backlog reasonably short, a request which has been in backlog for six months, will be declared non-viable and discarded. Discarded requests can be reopened in the future as priorities might change. Technical Director is allowed to pull back any request in to-do if more valuable requests show up. To-do WIP limit should be close to number of requests to be delivered before next Development meeting. Lead time calculation begins when request is elevated to To-do phase.

Actual design work starts when a design engineer pulls a request of her/his expertise area from to-do buffer according the predefined prioritization rules described in chapter 4.3.3. Requested parts are designed during design-in-progress phase. Each engineer has maximum of 3 available WIP slots for requests. Pair designing is encouraged, and peer reviewing should be done on every part designed for diverse approaches and higher quality designs. Swarming activity is used to remove impediments e.g. blocks. Finished designs need to have an assembly and part number generated, assembly in structure, peer and stakeholder review done and design and document package done. When design is finished it is moved into design-ready state and next item will be pulled from to-do. Technical Director will pull designs, review and approve design and mark design released for purchasing and move approved request to design-released buffer phase to wait purchasing. Design reviews should be done daily to ensure fluent flow of work. If part needs redesign, a request will be assigned to an engineer, who did the design for immediate corrections. A process issue is raised for further evaluation and to avoid unnecessary redesign work in the future. An active release approval state is omitted as it is very short. 30 minutes working time will be used as an estimate to calculate active working time.

Purchasing will pull requests from design-released phase where part number information has been automatically updated in to the Enterprise Resource Planning (ERP) system. Every purchaser may work simultaneously on maximum of three requests. Purchasing makes the final selection of supplier and creates purchase order (PO) in ERP. As order confirmation is received, it is added to PO and delivery estimate will be updated on request. Request will be moved to purchase-in-manufacturing state, during which the parts are manufactured and delivered by supplier. Purchasers will still follow deliveries and information from supplier and amend delivery date estimation if changes occur. As a part or first parts of assembly arrive to warehouse, a request will move into warehouse-received buffer phase. Quality control will pull items and validate them against specifications. Simultaneously workshop commits a test install. If both are fine, part gets quality control (QC) approval for use and request is put into warehouse-qc-approved buffer state. As specific active state is omitted, work time is estimated to be 30 minutes excluding test installation work.

Testing start from planning where test engineer pulls requests from warehouse-qc-approved phase. In test planning approval criterions are finalised, actual test schedule needed parts logistics planned. Request is moved to testing-planned phase to wait for actual tests. Work time is estimated to be 30 minutes per part or assembly, as specific active phase is omitted. A test engineer participating tests in certain test locations and conditions will pull all the requests to be tested to test-in-progress phase. Tests results are documented and linked into requests. If planned tests have been commenced for a requested part, it will be moved into testing-done phase. If additional tests are needed request is moved to testing-planned phase to wait for next suitable test session. The Technical Director will pull the requests from Testing-done and review the test results and approve part to be delivered by moving it to production-closed phase. If part needs redesign, Technical director has three reassign options. If opportunity window has been lost, discard request or move it back to backlog to wait for next opportunity or assign request back to design engineer who did original design. In every occasion a process issue is raised to further evaluation. Production approval phase is omitted and working time is estimated to be 30 minutes per request.

Two tasks, production planning and homologation have no separate process phase as they are done throughout the process phases from to-do to testing-done. Instead they have indicator on every request ticket which need to be checked after such work has been done. Active working time should be measured if efficiency meters are used.

There are three swimlanes in Kanban board to remark expedite and fixed date delivery priority classes. This way these high priority requests are raised to a separate space for easier discovery. Later, only ticket colour information could be used to visualise priority class and swimlanes could be used to indicate different work item types or part category to make selecting requests during pull easier to design engineers. Again, different ways should be experimented, and decisions made based on measured results.

4.3.5 Meet with the stakeholders about policies and coordination

During the third workshop goals, process, Kanban board, work item types and classes of service were finalized and approved. In addition to those three events: a development meeting, a daily standup meeting and an operations review meeting, was agreed.

Development meeting is held every Tuesday afternoon. Technical management team members are also members of development team and it is boarded by Technical Director. Development meeting is responsible of choosing backlog items to fill free slots in to-do.

A daily standup meeting is held every day at 10:30 (Eastern European Time) to accommodate UK members into meeting. Meeting is held on electronic kanban board for on-site members, combined with virtual meeting space for remote participants. Blocks, new issues and expedites are covered during the meeting. Also new entries in to-do phase and priority shifts can be introduced in Wednesday's meetings. If timeslot of 15 minutes allows, also progress from last meeting can be shortly reviewed. Everyone directly contributing to the process should participate and everyone else are free to join whenever they seem fit. Person currently working on request will shortly tell about it, but team should resist temptation to start troubleshooting the issue or diving in details to save everyone's time. Instead a swarming should happen after daily standup meeting by those persons who can help on matter at hands. Eventually, managements highest priority is to help removing any occurring impediments.

Operations review meeting is held every 3–4 weeks, after the product shipment to retrospectively evaluate results. Again, everyone directly contributing to the process should participate. Operations Director and Technical Director will go through results from latest product shipments and tests. Also, last period process results are presented and process development tasks (issues) and those progress is covered. Optionally, a process development workshop (retrospective) is run to identify new development ideas and assign responsible person for those.

4.3.6 Create an electronic board

Microsoft Azure Devops was chosen as an electronic Kanban board by DMA team as it integrates nicely to other IT systems they use and as they are already partners with Microsoft. Azure Devops integrates easily to Microsoft Teams, which is a primary communication channel used in virtual meetings, team and bilateral communication, a document system and a discussion board. Microsoft engineering services build the initial electronic board, which was iterated during 5 different sessions and at the same time a knowledge transfer of building and managing basic functions was conducted, so that DMA could keep up the board by themselves after launch. A sample of an electronic board is in Appendix 8.

During the iterations also the request content and parameters, which had subset of acceptable values, were defined. All the items in current excels were reviewed by Technical director and successive tasks were imported in to Azure Devops just before trainings and launch.

4.3.7 Educate the team on the new board

A planning phase of research ended to a training day in DMA premises. Two 1,5 hours training sessions were conducted, both with similar agenda. Agenda covered three 30 minutes topics: a goal setting, a process walkthrough and a demo of using electronic Kanban board.
Around 30 persons attended to these two sessions. Dozen questions were raised and answered during the sessions and discussion was open and lively. During the sessions it was underlined, that initial approach follows as much as possible how things are done today, but it adds visibility and clear approval points and criterions. Having a 15 minutes daily meeting raised some protests, but participants came into common agreement to test it for a one week and after that decide whether to decrease the number of events during the week.

A tool was considered easy to use and flexible, so that everyone could get alerts either to email or Teams channel. People was also shown how they could use system for discussions of certain requests and how rich content or links to external systems could be used to collect necessary information in one place.

4.4 **Experimentation**

Experimentation of new methodology started on the following day from team training. A first milestone was after a week of having daily standup meetings. Team somewhat unanimously decided to continue this practice, as it was increasing information exchange between the participants and saved time as need for having one-to-one discussions decreased. Participant numbers were estimated during dailies and number of attendants varied between 14–20 during nine-week experimentation period. Researcher participated remotely, with Microsoft Teams collaboration application, to 2–3 dailies per week, giving feedback and improvement suggestions weekly.

A summary of observations and feedback:

During experimentation, only few people participated remotely. Remote participation is likely to increase if new methodology will be brought into play also in remote teams. In such situation it is good to remember that fluent virtual meetings require high quality audio microphone capturing voice of farther positioned participants in the room. Other option is to use several boundary microphones to cover whole area. Having participants positioned around microphone helps also issue. Only one person should be speaking at the time, otherwise it gets a lot harder to follow discussion. Having live video between the participants increase interaction and helps to identify people who are talking. Screen sharing was used from beginning and it worked flawlessly. Azure Devops has possibility to visualize task changes in real time, but it is not as immersive as screen sharing as other participants cannot see view changes or mouse pointer.

Discussions prolonged in several occasions. If issue concerns all the participants, it is not a problem. But if discussion concerns only few participants, rest of the team could feel redundant. Primary goal is to identify the issue and participants, who can assist fixing it. Same applies to process improvement suggestions. Issue can be raised and assigned, but corrective measures should be planned in separate session.

Some of the agreed information on tickets were fulfilled sparingly, leaving contributors downstream in the process without required information. It will also make documentation incomplete if there is a need to do analysis later. In general, every action, information or definition of done should be justified and agreed by the process contributors. By definition, these explicit rules must be followed, or rules needs to be changed.

As small requests to increase usability of process or tool raised, those were implemented in short notice. This increases perception that opinions matter and encourages everyone to contribute. Greater changes require more planning and possibly evaluation and comments from affected persons. Anyway, experimentation of process changes should be embraced and measured.

During time periods between tests and product shipments, there was usually high amount of expedites in the system. This is likely because of the nature of operation, but also related to having issues with lead time predictability and prioritisation. As maturity increases, amount of expedites should be limited as they are likely to break the process flow.

Some of the daily standup meetings were arranged in a meeting room. This has two downsides. When people are allowed to sit down, it tends to prolong the meetings. Also having DSM’s in open space can draw occasional people to get a short update of team’s progress.

After three weeks, there was a suggestion to decrease some of the WIP limits. Based on measurements it should be safe to reduce To-do to 25, Design in progress to 18, Design ready to 9, Design released to 9, In manufacturing to 40, Warehouse received to 10, Warehouse QC approved to 10, Testing planned to 25, Test in progress to 75 and Testing approved to 75. Rationale behind this is to get bottlenecks visible and possibly smaller amount of unfinished work within the system and eventually shorter lead time. Limits could be reduced even more aggressively.

In the end of the experimentation period, there was 93 unfinished assignments in the system between Design-in-progress and Testing-Approved. If during last couple of years amount of finished assignments has been in between 8–12 items per week, this seems quite high but bearable amount. Reducing WIP limits would help decreasing amount of unfinished tasks.

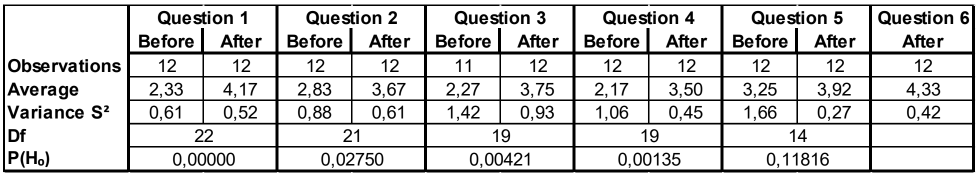
Only real-time report that was available during the experimentation period, was cumulative flow diagram. Other suggested reports included: a control chart illustrating daily mean lead time, lead time variance, weekly rolling lead time and scatter chart to identify outlier assignments; lead time and due date performance table; issues and blocked items chart; and velocity/team happiness diagram.

Participation to events and discussions was active and in positive atmosphere. In some of the events I would call it even enthusiastic. A lot of positive and constructive feedback was given during the period and some great improvement ideas were raised, which were not possible to implement during this experimentation because of time and resource constraints. One great improvement was that team decided to label every task according test or product shipment, which helped following whether these parts can make it to the scheduled test or shipment.

Unfortunately, one of the planned events, a production meeting, was reduced in number of participants and scope because of aforementioned constraints. Despite that several impactful improvement tasks were identified related to splitting tasks before moving those in To-do, preliminary production planning, assignment documentation, approval of parts to shipped products use and returning partial assignments back for redesign. As production meeting includes retrospective, it should be acknowledged as a key event for continuous improvement. I warmly encourage the team to experiment with an all-inclusive production meeting event after couple of days of every batch shipment or within four weeks of previous production meeting. As a half-day session, it takes significant amount of time, but debriefing the latest shipped batch, reviewing product development process measurements including previous period issues and blocks, previous period improvement actions and a retrospective session by using Spotify Retro-kit (Österberg, Esni, Rabiee & Majkowska 2017) as an example to identify new improvement activities.

4.5 Evaluation

As described in Chapter 3.3 two rounds of surveys were commenced to validate experimentation results. Both surveys were sent to 58 persons. Only 12 persons participated to both rounds, which makes response ratio quite small, although most if not all, respondents are likely direct process contributors. Response arithmetic means, variance and T-test were calculated for each of the question pairs. Results are in Table 5.



	Question 1		Question 2		Question 3		Question 4		Question 5		Question 6
	Before	After	Before	After	Before	After	Before	After	Before	After	After
Observations	12	12	12	12	11	12	12	12	12	12	12
Average	2,33	4,17	2,83	3,67	2,27	3,75	2,17	3,50	3,25	3,92	4,33
Variance S^2	0,61	0,52	0,88	0,61	1,42	0,93	1,06	0,45	1,66	0,27	0,42
Df	22		21		19		19		14		
$P(H_o)$	0,00000		0,02750		0,00421		0,00135		0,11816		

Table 5. Two-round web-based survey results.

Statement in question one was “Product Development Process is clear and easy to follow”. Results mean increased during experimentation from 2,33 to 4,17 results being statistically highly significant as probability for null hypothesis ($P(H_o)$) is less than 0,01.

Statement two was “Product Development Process supports design and production of quality components”. Results mean improved from 2,83 to 3,67 and those were statistically significant as $P(H_o)$ was between 0,01 and 0,05.

Statement three in surveys were “Request prioritisation in Product Development Process is clear and helps to focus to most important requests”. Response mean raised from 2,27 to 3,75 with statistically high significance.

Statement four “Product Development Process provides feedback about request status and results” was related to communication. Results improved from 2,17 to 3,5 being statistically highly significant.

Statement five, “I am able to give my best effort to support our team’s success” result was only one without statistical significance as $P(H_o)$ was 0,118 which is more than 0,05. For this question, a response mean increased from 3,25 to 3,92.

Statement six was used only during the second round, as it surveyed perception related to experimentation. Mean score to statement “Changes have improved the Product Development Process” was 4,33 with variance of 0,42. Figure 23 visualises survey results.

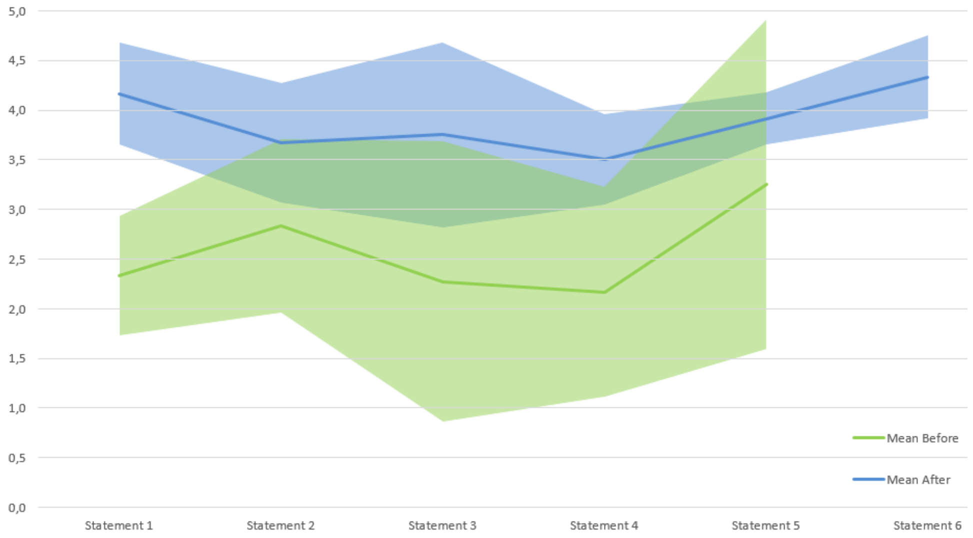


Figure 23. Response means and variances for survey statements.

During initial survey round, there was no free form feedback. Fortunately, on the second round, five respondents gave comments related to following topics grouped by similar themes:

- *Implementation of Kanban methodology is giant leap forward, but we can still go further.*
- *We need bigger board, someone to read topics aloud or otherwise highlight topics in discussion for easier following of dailies.*
- *Kanban board and process itself is good, but we have to start use it properly.*
- *We have now tools to control items going for design.*
- *We need to work with prioritisation and ticket content quality in the beginning of the process.*
- *Integrating part numbering better into process and tools could help reassigning tasks or redesigning parts. As team grows this becomes more important.*
- *We still need operations handbook to describe our ways of working and expected quality to induct new team members.*
- *Release process and signoff's need to be clarified.*
- *We need to find ways to close/approve items effectively.*
- *We need to define more clearly reviews and approvals of design phase.*
- *We don't respect approvals and signoff's strictly enough, and lot of important information is missing, especially in the beginning of the process.*
- *Very positive participation from all the participants.*

Many of these comments are related to improvement opportunities recognized in the Operations Review meeting. Some of these can be improved with marginal effort, just by identifying and steering actions which does not comply with agreed process rules. Some require operating model finetuning, additional instructions or definitions for clarification. Some are more challenging or require longer time, as changes to business applications may prove to be laborious.

As there was a desire to find process productivity information related to the change, the number of releases per week was calculated based on product lifecycle management system information. According to this data a year-over-year comparison chart was created in Figure 24.

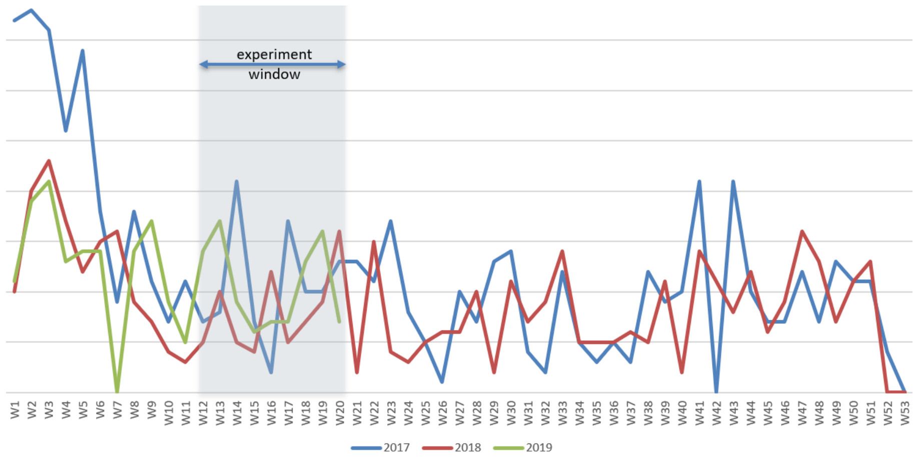


Figure 24. Year-over-year comparison of design releases.

Also, a mean for full period and 12 weeks rolling average was calculated (Figure 25). But neither of these indicate significant visible change in results during experimentations. Even though experimentation period release mean increased notably from same periods in 2018 (+31,5 %), it was only subtly higher that same period in 2017 (+1,1 %) or the full calculation period mean (+1,6 %). Also compared to preceding nine-week period, productivity was down (-10,3 %). These variations are also caused by amount of available resources, variance in work item sizes and delivery schedules and thus productivity gains or losses remain inconclusive.

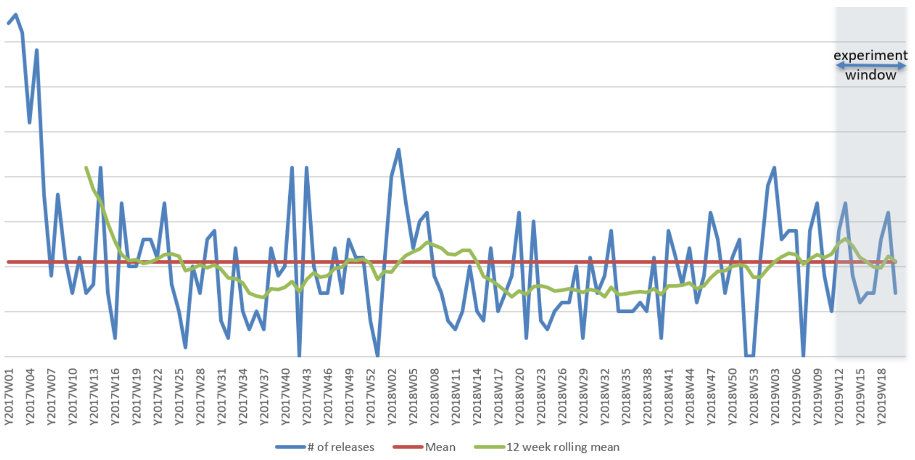


Figure 25. Full calculation period results with total and 12 week means.

As a last measurement of result of the changes, DMA management gave following project feedback:

"This research gave us very valuable kick off for process and operations development and this is visible in the survey results as well. The research provided a model which suits our environment and enables the continuous improvement to be done ourselves. Very crucial part of the research to succeed were the initial interviews to identify the pressure points but also the researcher's active role during implementation by giving guidance and regular feedback. The initial step with the thesis is a major step forward but most important result for us is the changed culture. Pressure points in the process are now discussed when seen and we actively change procedures to seek improvement."

5 CONCLUSION

As described in Chapter 3.1, the research problem was to understand whether Agile methodologies could provide benefits in Product Development Process of DMA.

The research questions were:

- is there need to change the DMA Product Development Process
- could Agile software development methodologies be applied
- which Agile development framework fits best to DMA needs
- what kind of results (improvement) new operating model provides

Based on semi-structured interviews, a need for process changes were evident. Interviews provided rich description of current status and simultaneously raised most pressing issues of current operating model. Improvement was clearly needed on communication, prioritization and having better structured development process. To ensure reliability, interview process has been documented, so that it is possible to re-run interviews, but as the result of this research, situation has changed and being able to receive same results is not possible. Interview material has been collected by following best practices on interviewing, recording, transcription and analysis of material. Validity has been ensured by having a sample size in which a saturation has clearly formed and verifying analysis result with semantic analysis tools.

Based on literature and available media, agile and lean methodologies are increasingly used, not only in ICT industry, but also in manufacturing superseding traditional waterfall and stage-gate approaches. Lean manufacturing having significant footprint in automotive industry was a compelling approach, but at the same time continuous and incremental development of highly customised products has many common challenges with modern software development. During exploration of reference material, it become apparent that some of the methodologies adopted by rapidly increasing ICT technology companies could be translated to context of research subject.

Most prominent Agile methodologies were studied during this research and based on those studies, in combination of research subject specific requirements, Kanban was chosen as most fit-for-purpose methodology. Evaluation areas were documented and rationale of results for every assessed methodology. Reason for this choice was primarily Kanban's non-intrusiveness, flexibility, active development and available reference material. To increase reliability researcher provided the information and DMA team did definitive decision which methodology will be used.

It was already given in the beginning, that if process will be changed, researcher would have an active role in planning and implementation. As Anderson (2010, 167–176) provide a detail description of how to implement Kanban, it was used as a template, and most time-consuming task was creation of virtual kanban board according planned process traits. Also, implementation of the new operating model was quite straightforward as it was following earlier process, but introducing practices for prioritisation, visualisation of status and feedback loops to increase communication.

Based on identical surveys before and after the change, persons contributing to process perceive new approach to be significant improvement compared to former process. These surveys are documented so that they are rearrangeable, but as way of working evolves, having same results becomes unlikely. Results has been evaluated by following statistical analysis tools used in quantitative research. Effectiveness of the process could be subtly better or at the same level as before, which is an acceptable result, as operating model changes can introduce short time interval negative impact to productivity.

One of the most significant change is reflected from the DMA representative’s comment that “during the past two months we have been discussing more how we could improve the development process than past two years”, which illustrates increased continuous improvement activities.

Further studies could be done by adopting tools used in this research to verify, whether similar results can be achieved with other subjects. As there are already quite many non-ICT single organisation case studies of agile methodologies use available, a meta study of researches could help to identify whether agile methodologies provide improved results over more traditional approaches. And what are the context specific conditions which indicate improved results. Additionally, having more longitudinal study with same research subjects would provide information about possible long-time productivity and employee satisfaction improvements and how well continuous improvement has been established into these organisations.

At the same time, I feel that without constraints related available time and resources of both DMA team and myself, we could have been able to provide even more significant improvement. Having production reviews with whole team more frequently and all planned measurements and reporting in place, would have made additional significant positive impact. Due to my opinion based on empirical experience during this research and my career, the most important practice to master is continuous improvement. If you nail it, it doesn't matter what is your current position, as you are evolving faster than your competition.

Based on these results I judge this project to be more of a step than a leap to right direction. Just like a single iteration in Agile development. I hope that the great, highly skilled individuals in DMA will continue on this path, study Lean and Agile, experiment open-mindedly new practices and measure constantly progress for decision making and embraces inclusion of whole team contributing to results.

> *"Stop trying to borrow wisdom and think for yourself. Face your difficulties and think and think and think and solve your problems yourself."*
>
> *—Taiichi Ohno*

REFERENCES

Agile Alliance (2018). What is Agile Software Development? Retrieved at 7.11.2018 from https://www.agilealliance.org/agile101/

Anttila, P. (2006). *Tutkiva toiminta ja ilmaisu, teos, tekeminen.* Hamina: Akatiimi Oy.

Manifesto for Agile Software Development. (n.d.). Retrieved at 7.11.2018 from http://agilemanifesto.org/

Anderson, D. J. (2010). *Kanban: Successful evolutionary change in your software business*. Sequim, WA: Blue Hole Press.

Aukia, P., Luoto, K. & Tiainen, M. (2017, October 27). Selvitys: Lean-menetelmät Suomessa 2016. Retrieved at 6.9.2018 from https://www.codento.fi/2016/04/selvitys-lean-menetelmat-suomessa/

Aukia, P., Kuha, M., Luoto, K. & Tiainen, M. (2017, May 3). Selvitys: Lean-menetelmillä parempaa kilpailukykyä muutoksessa. Retrieved at 6.9.2018 from https://www.codento.fi/2017/05/lean-selvitys-2017/

Aukia, P., Iivonen, J. & Luoto, K. (2018, June 21). Raportti: Lean-menetelmät Suomessa 2018. Retrieved at 6.9.2018 from https://www.codento.fi/2018/06/raportti-lean-menetelmat-suomessa-2018/

Beck, K. & Andres, C. (2005). *Extreme programming explained: Second edition, embrace change*. Boston: Addison-Wesley.

Blondel, V. D., Guillaume, J., Lambiotte, R. & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment, 2008*(10).

Brown, A. & Justice J. (2018). Scrum for Hardware: Full Scale Manufacturing. Retrieved at 12.3.2019 from https://www.scruminc.com/scrum-hardware-full-scale-manufacturing/

CA (2018, March 29). Survey Data Shows That Many Companies Are Still Not Truly Agile. Retrieved at 7.4.2019 from https://hbr.org/sponsored/2018/03/survey-data-shows-that-many-companies-are-still-not-truly-agile

Florian, M. (2016). There is no Spotify model. Retrieved at 27.1.2019 from https://www.infoq.com/presentations/spotify-culture-stc

Gladwell, M. (2000). *The Tipping Point: How Little Things Can Make a Big Difference.* Boston (MA): Little, Brown and Company.

Justice, J. (2011). WIKISPEED Keynote 5/18/11 (1 of 2) - Using Agile, Lean and Scrum. Retrieved at 12.3.2019 from https://www.youtube.com/watch?v=CNhdfAxa648

Justice, J. (2011). WIKISPEED Keynote 5/18/11 (2 of 2) - Using Agile, Lean and Scrum. Retrieved at 12.3.2019 from https://www.youtube.com/watch?v=M7Uv33fOLXA

Kananen, J. (2014). *Toimintatutkimus kehittämistutkimuksen muotona*. Jyväskylä: Suomen Yliopistopaino Oy.

Kniberg, H. & Skarin, M. (2010). *Kanban and Scrum: Making the most of both*. S.l.: C4Media.

Kniberg, H. & Ivarsson, A. (2012 October). Scaling Agile @ Spotify with Tribes, Squads, Chapters & Guilds. Retrieved at 27.1.2019 from https://blog.crisp.se/wp-content/uploads/2012/11/SpotifyScaling.pdf

Kniberg, H. (2014, March 27). Spotify engineering culture (part 1). Retrieved at 27.1.2019 from https://labs.spotify.com/2014/03/27/spotify-engineering-culture-part-1/

Kniberg, H. (2014, September 20). Spotify engineering culture (part 2). Retrieved at 27.1.2019 from https://labs.spotify.com/2014/09/20/spotify-engineering-culture-part-2/

Ladas, C. (2008). *Scrumban: And other essays on kanban systems for lean software develoment*. Seattle (WA): Modus Cooperandi Press.

Larman, C. & Basili, V. (2003). Iterative and incremental development: A brief history. *Computer, 36*(6), 47–56.

Leavitt, H. (1972). *Managerial Psychology (3rd edition)*. Chicago: The University of Chicago Press.

Liker, J. K. (2010). *Toyotan tapaan*. Helsinki: Readme.fi.

Morgan, J. M. & Liker, J. K. (2006). *The Toyota Product Development System: integrating people, process and technology.* New York: Productivity Press.

Mäkelä, K. (1990). *Kvalitatiivisen aineiston analyysi ja tulkinta*. Helsinki: Gaudeamus.

New York Times (2011, October 2). Books | Best Sellers. Hardcover Advice & Misc. Retrieved at 23.2.2019 from https://www.nytimes.com/books/best-sellers/2011/10/02/hardcover-advice/

Niiniluoto, I. (1984). *Tiede, filosofia ja* maailmankatsomus. Helsinki: Otava.

Reifer, D. J. (2017, August 10). Quantitative Analysis of Agile Methods Study (2017): Twelve Major Findings. Retrieved at 7.4.2019 from https://www.infoq.com/articles/reifer-agile-study-2017

Ries, E. (2011). *The Lean Startup. How constant innovation creates radically successful businesses.* London: Portfolio Penguin.

Schwartz, F. (2017). Tangible Scrum. Retrieved at 10.3.2019 from https://www.scrumalliance.org/ScrumRedesignDEVSite/media/scrumalliancemedia/files%20and%20pdfs/community/webinars/agile%20leadership/al_presentation_06212017.pdf

Scrum Org. (n.d.). What is Scrum? Retrieved at 16.9.2018 from https://www.scrum.org/resources/what-is-scrum

Scrum Inc. (n.d.). The Scrum in Hardware Guide. Retrieved at 12.3.2019 from https://www.scruminc.com/scrum-in-hardware-guide/

Serrador, P., Pinto, J. K., (2015). Does Agile work? – A quantitative analysis of agile project success. *International Journal of Project Management, 2015*(33).

Sutherland, J. (2015). *Scrum: The art of doing twice the work in half the time*. London: Random House Business Books.

Sutherland, J. & Schwaber, K. (2017). The Scrum Guide. Retrieved at 11.9.2018 from https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf

Takeuhci, H. & Nonaka, I. (1986). The new new product development game. *Harvard Business Review 64, January-February 1986*, 137–146.

Versionone (n.d.). State of Agile Report. Retrieved at 11.9.2018 from https://explore.versionone.com/state-of-agile

Wikispeed (n.d.). Who we are. Retrieved at 12.3.2019 from http://wikispeed.org/about/

Österberg M., Esni B., Rabiee S., Majkowska Z. (2017, December 15). Spotify Retro Kit. Retrieved at 22.4.2019 from https://labs.spotify.com/2017/12/15/spotify-retro-kit/

Appendix 1

RESEARCH TIMELINE

#	Phase description	Schedule
1	Survey of current operating model	W38–42
1.1	Gathering current documentation • Basic information about company • Organization and steering model • Processes • Tools and technology • Measuring results	W38–40
1.2	Workshop • Current operating model walkthrough	W39 27.9.2018
1.3	Interviews • Current operating model o Strengths and weaknesses • Need for change • Identification of objectives for change • Preliminary information to support selection of target operating model framework • Transcription	W40–44 2.10.2018 15.10.2018 26.10.2018 30.10.2018
2	Analysis and affecting factors	
2.1	Interview data categorization and interpretation	W42–44
2.2	Problem synthesis based on material	W43–45
2.3	Synthesis walkthrough and follow-up proposal • Interview findings • Proposal of future operating model framework	W45 7.11.2019
3	Target operating model planning	W43–45
3.1	Preliminary planning of target operating model	W43–45
3.2	Info session about Agile frameworks	W39 27.9.2018
3.2	Planning workshop • Adapting framework to context • Processes, roles, tools and measurements	W45 7.11.2018
3.3	Implementation workshop • Schedule • Communication	W51 21.12.2018
3.4	Target operating model walkthrough and acceptance	W05 29.1.2019
4	Implementation of target operating model	
4.1	Measurement of current operating model	W10–11
4.2	Tools implementation	W51–08
4.3	Training	W11 14.3.2019
4.4	Communication	W11 14.3.2019

4.5	Go-live	W11 15.3.2019
5	Check results	
5.1	Verify target operating model is followed	W12–20
5.2	Verifying measurement results	W12–20
5.3	Measurement of Target Operating Model	W20
5.4	Finishing documentation	W21–24
5.5	Documentation commenting	W23–25
5.6	Results and feedback session	W26

QUESTIONS FOR SEMI-STRUCTURED INTERVIEW (V1.0)

Theme 1: Current state of R&D operating model

- Describe current R&D operating model.
- Are you following process as it is documented?
  - If not, how does actual process differentiate from documentation?
- How tasks are raised to backlog?
- How tasks are being approved or prioritized in backlog?
- Do you have a comprehensive list of tasks in backlog?
  - Who have access to that list?
- How is R&D process managed/steered?
- Can you identify R&D process stakeholders?
- How R&D team is exchanging information and knowledge internally?
- How R&D team is exchanging information with stakeholders?
- How finished requests are being approved (hand over)?
- Describe how are you developing your work methods.

Theme 2: Need for change

- What are the strengths of current operating model?
- What needs to be changed in current operating model?
- What are the impediments preventing you to achieve faster lead time?

Theme 3: Organizations ability and willingness to change

- If operating model were changed, what would you see as limiting factors for new model?
  - What things need to be considered?
  - What things cannot or should not be changed?
- Do you think that your team is open or willing to change?

QUESTIONS FOR SEMI-STRUCTURED INTERVIEW (V1.1)

Theme 1: Current state of R&D operating model

- Describe current R&D operating model (from raising an issue or request to delivering a product).
- Are you following process as it is documented?
- What are the objectives of R&D process?
- How issues or requests are raised to backlog?
- How tasks are being approved or prioritized in backlog?
- Do you have a comprehensive list of tasks in backlog?
- How is R&D process managed/steered?
- Can you identify R&D process stakeholders?
- What common tools or systems are used to manage R&D process
- How R&D team is exchanging information and knowledge internally?
- How R&D team is exchanging information with other stakeholders?
- How finished requests are being approved?
- Describe how are you developing your work methods.
- Describe utilisation or resourcing level of R&D personnel.

Theme 2: Need for change

- What are the strengths of current operating model?
- What needs to be changed in current operating model?
- What are the impediments preventing you to achieve objectives?

Theme 3: Organizations ability and willingness to change

- If operating model were changed, what would you see as limiting factors for new model?
  - What things need to be considered?
  - What things cannot or should not be changed?
- Do you think that your team is open or willing to change?

Appendix 3

INVITATION TO INTERVIEW

PURPOSE OF THE INTERVIEW

Theme (semi-structured) interviews of R&D team and process stakeholders are part of the research about benefits of agile methodologies in Discrete Manufacturing and Assembly Company's Research and Development process. Purpose of this interview is to understand current state of R&D operating model, evaluate the need and potential benefits of the change, and survey possible limiting factors for the change. Interviews are one of the primary information sources for the research.

HOW ARE INTERVIEWS CONDUCTED

Participation to interview is voluntary. Interviewee has right to refuse to answer to any of the questions and suspend his/her participation. Interviews are recorded, translated (if needed), transcribed in standard language, and anonymized by possible individual references, pseudonymized by the respondent, after which the identification of individual interviewees is not possible. Original recordings are properly deleted. Role and organization, they represent will be published in the thesis report. The thesis report is published in the open Theseus database. Interviewee has right to obtain further information from researcher about the research at any time during the research.

By participating in the interview, the interviewee will give permission to use the information collected during the interview, as described above, unless specifically prohibited by interviewee.

Appendix 4

INTERVIEW TOPIC CATEGORY CROSS-REFERENCE HEAT MAP

# of OCC	TOPIC	Communication	Management	Process	Tools	Documentation	Inclusion	Validation	Schedule	Change	Objectives	Assignment	Defects	Priorisation	Resourcing	Review	Improvement	Partners	Knowledge	Roles	Time-pressure	Impediments	Agility	Requirements	Homologation	Resistance	Iteration	Benefits	Front-loading	Completion	Initiative	# of XREF
130	Communication		17	11	22	13	21	17	5	21	12	4	11	1	1	2	4	9	2	1	0	1	1	2	0	3	1	0	0	1	1	184
127	Management	17		28	8	4	8	20	18	8	14	12	2	13	1	13	1	1	2	17	0	3	2	2	1	0	0	0	1	2	1	199
113	Process	11	28		16	11	4	6	14	14	14	8	12	4	2	7	12	0	1	10	2	0	4	6	4	3	0	0	0	1	0	194
105	Tools	22	8	16		28	3	9	6	14	3	12	12	2	2	3	2	1	0	1	0	1	2	1	6	1	0	0	0	0	1	156
81	Documentation	13	4	11	28		9	15	1	9	1	6	8	1	2	0	0	3	11	1	1	0	2	2	4	0	5	0	1	1	0	139
75	Inclusion	21	8	4	3	9		2	5	10	4	1	1	1	2	8	6	9	10	0	1	1	0	3	2	1	0	2	5	0	1	120
75	Validation	17	20	6	9	15	2		5	6	9	4	13	0	1	4	2	5	3	2	1	0	0	6	2	1	0	0	1	1	0	135
72	Schedule	5	18	14	6	1	5	5		7	9	9	0	11	2	4	3	2	2	3	8	2	1	4	5	0	0	1	4	1	0	132
71	Change	21	8	14	14	9	10	6	7		12	3	2	1	4	3	6	2	4	4	0	0	1	2	1	0	0	0	4	1	0	139
66	Objectives	12	14	14	3	1	4	9	9	12		3	1	6	1	2	1	0	1	2	2	2	2	7	1	3	0	1	1	0	0	114
54	Assignment	4	12	8	12	6	1	4	9	3	3		5	8	0	8	0	0	0	1	0	0	0	2	0	0	2	0	0	0	1	89
46	Defects	11	2	12	12	8	1	13	0	2	1	5		1	1	2	0	3	3	0	2	0	2	3	0	0	2	0	0	0	1	87
42	Priorisation	1	13	4	2	1	1	0	11	1	6	8	1		0	7	1	0	1	0	3	0	0	1	1	0	0	0	0	2	0	65
36	Resourcing	1	1	2	2	2	2	1	2	4	1	0	1	0		0	3	1	1	2	2	9	0	0	0	0	0	4	0	1	0	42
36	Review	2	13	7	3	0	8	4	4	3	2	8	2	7	0		1	0	0	1	0	1	0	3	0	1	0	0	2	0	0	72
34	Improvement	4	1	12	2	0	6	2	3	6	1	0	0	1	3	1		0	0	4	0	0	1	1	0	1	0	0	1	1	1	52
32	Partners	9	1	0	1	3	9	5	2	2	0	0	3	0	1	0	0		6	0	0	0	0	3	1	0	1	1	3	0	0	51
29	Knowledge	2	2	1	0	11	10	3	2	4	1	0	3	1	1	0	0	6		0	0	0	0	2	1	0	0	1	6	0	1	58
29	Roles	1	17	10	1	1	0	2	3	4	2	1	0	0	2	1	4	0	0		0	0	0	0	0	2	0	1	0	1	0	53
24	Time-pressure	0	0	2	0	1	1	1	8	0	2	0	2	3	2	0	0	0	0	0		1	1	0	0	0	0	0	0	1	0	25
23	Impediments	1	3	0	1	0	1	0	2	0	2	0	0	0	9	1	0	0	0	0	1		0	0	1	0	0	0	0	0	1	23
22	Agility	1	2	4	2	2	0	0	1	1	2	0	2	0	0	0	1	0	0	0	1	0		0	0	0	0	0	0	2	1	22
21	Requirements	2	2	6	1	2	3	6	4	2	7	2	3	1	0	3	1	3	2	0	0	0	0		2	1	0	0	1	0	0	54
18	Homologation	0	1	4	6	4	2	2	5	1	1	0	0	1	0	0	0	1	1	0	0	1	0	2		0	0	0	0	0	0	32
17	Resistance	3	0	3	1	0	1	1	0	0	3	0	0	0	0	1	1	0	0	2	0	0	0	1	0		0	0	0	0	0	17
17	Iteration	1	0	0	0	5	0	0	0	0	0	2	2	0	0	0	0	1	0	0	0	0	0	0	0	0		2	1	1	0	15
16	Benefits	0	0	0	0	0	2	0	1	0	1	0	0	0	4	0	0	1	1	1	0	0	0	0	0	0	2		1	1	0	15
15	Front-loading	0	1	0	0	1	5	1	4	4	1	0	0	0	0	2	1	3	6	0	0	0	0	1	0	0	1	1		0	0	32
13	Completion	1	2	1	0	1	0	1	1	1	0	0	0	2	1	0	1	0	0	1	1	0	2	0	0	0	1	1	0		0	18
7	Initiative	1	1	0	1	0	1	0	0	0	0	1	1	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0		10

Appendix 5

EVALUATION CHART OF AGILE METHODOLOGIES

Methodology	Process changes	Role changes	Changing priorities	Suitability for size	Suitability for purpose	Popularity	Available documentation	Active community	Score
Scrum	2	2	2	2	3	3	3	3	20
Extreme Programming	1	1	2	2	1	1	3	1	12
Kanban	3	3	3	3	3	2	2	3	22
Scrumban	2	3	2	3	3	1	2	2	18
Lean Startup	2	3	3	3	1	2	2	3	19
Spotify Model	2	2	3	3	2	1	1	2	16
Extreme Manufacturing	2	2	2	2	3	1	1	1	14

Appendix 6

KANBAN BOARD TEMPLATE

	Backlog Mgmt (∞)		Design (50)			Purchase (160)				Test (160)			Prod. (∞)
	Backlog 6 months	To-Do 20/30	In Progress 21/6	Ready 5/10	Released 5/10	RFQ 3/3	Delivery 40/40	Received 4/16	QC Approved 10/40	Planned 10/40	In Progress 30/120	Done 30/120	Closed MTD + LM
Expedite													
Fixed Delivery Date													
Standard													

DoD:
Rough production plan
Person for feedback and process

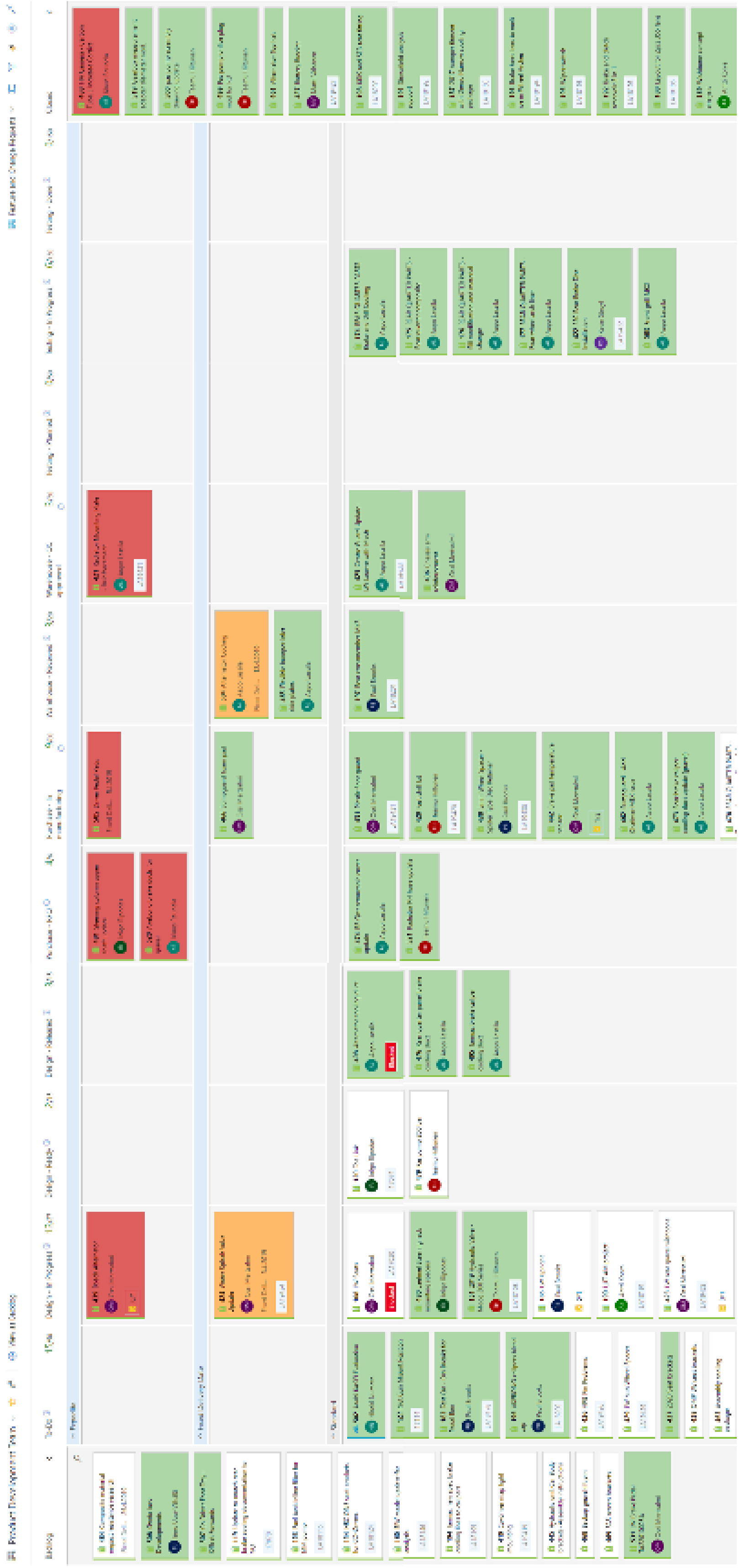
DoD:
(IP) Assembly part number generated
(IP) Assembly/part in structure
(IP) Peer and stakeholder review done
(IP) Desing and document package ready
(RY) Review completed and approved
(RY) Design released for Purchasing
(RD) Part number info updated in CRM

DoD:
(RFQ) Supplier selected
(RFQ) PO created in ERP
(RFQ) Order confirmation received and added to PO
(D) First items received to warehouse
(R) Items checked against specification
(R) Test installation completed
(QC) Approval criteria finalize
(QC) Testing planned
(QC) Needed part logistics planned

DoD:
(P) Part or assembly is under testing
(IP) Test completed and report available
(D) Decision made if part is ok shipping, redesign needed or idea is scrapped

Appendix 7

ELECTRONIC KANBAN BOARD



Appendix 8/1

INVITATION E-MAILS TO WEB-SURVEYS

Before the change survey invitation letter:

Welcome to the survey!

As you already know we have been preparing changes to Product Development Process in Discrete Manufacturing and Assembly Company. Kanban embraces scientific experimentation and that's why we want you to give your honest opinion about the changes. To be able to measure change, we will run two surveys. One before new process is being implemented and another after new process has been in use for a couple of weeks.

This survey will consist of five statements and you are asked to assess your level of agreement on scale from "Strongly disagree" to "Strongly agree". You can also skip a question by selecting "No answer". As a last question, there is also possibility to give freeform feedback. This will only take about five minutes. You'll find your personal link to the survey in the end of this e-mail and you are able to give your feedback until 13.3.2019 (EOD).

This survey is executed in Webropol, which is one of the biggest online survey companies in Nordic. Survey is done fully anonymously, meaning that only your e-mail address has been added to system to send you a personal invitation link, but that personal information will not be available in any reports. If you have any questions, please send e-mail to tero.lappalainen@student.hamk.fi. Do not use reply-to as this message has been send from survey platform and e-mail replies cannot be processed.

Thank you for your input!

Best Regards
Tero Lappalainen

https://link.webropolsurveys.com/R/ΨΨREDIRECTION_LINKΦΦ

After the change survey invitation letter:

Welcome to the follow-up survey!

You have now been experimenting with new Product Development Process for two months. It is time to evaluate the impact of the change by asking your honest opinions. As you probably remember, this is a second survey round. If you missed the first round, please don't let that prevent you to give your response this time.

This follow-up survey consists of same statements as the first survey and additional statement related to changes. You are asked to assess your level of agreement on scale from "Strongly disagree" to "Strongly agree". You can also skip a question by selecting "No answer". As a last question, there is also possibility to give freeform feedback. This will only take about five minutes. You'll find your personal link to the survey in the end of this e-mail and you are able to give your feedback until 16.5.2019 (EOD).

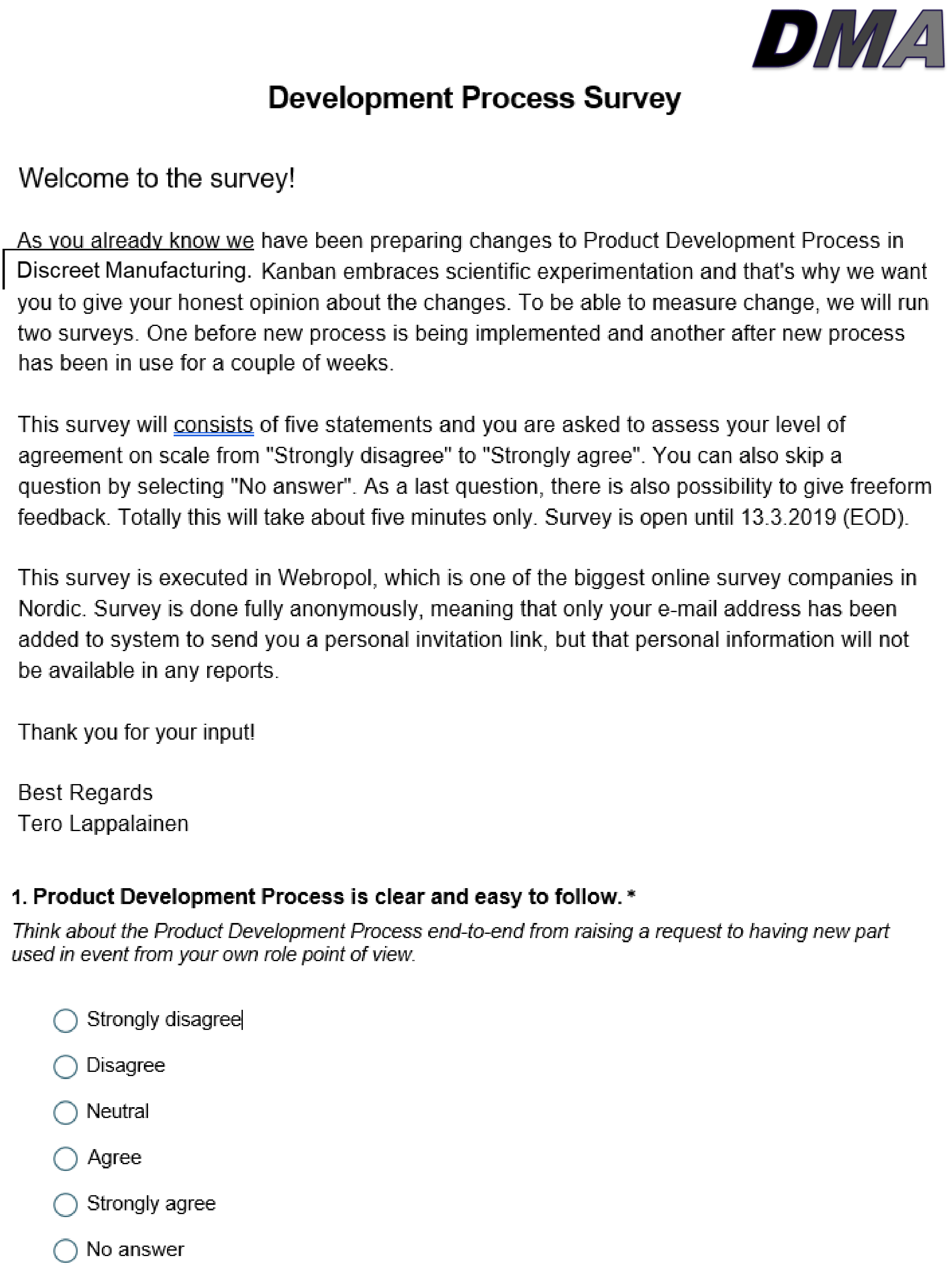
This survey is executed in Webropol, which is one of the biggest online survey companies in Nordic. Survey is done fully anonymously, meaning that only your e-mail address has been added to system to send you a personal invitation link, but that personal information will not be available in any reports. If you have any questions, please send e-mail to tero.lappalainen@student.hamk.fi. Do not use reply-to as this message has been send from survey platform and e-mail replies cannot be processed.

Thank you for your input!

Best Regards
Tero Lappalainen

https://link.webropolsurveys.com/R/ΨΨREDIRECTION_LINKΦΦ

WEB SURVEY QUESTIONS



Development Process Survey

Welcome to the survey!

As you already know we have been preparing changes to Product Development Process in Discreet Manufacturing. Kanban embraces scientific experimentation and that's why we want you to give your honest opinion about the changes. To be able to measure change, we will run two surveys. One before new process is being implemented and another after new process has been in use for a couple of weeks.

This survey will consists of five statements and you are asked to assess your level of agreement on scale from "Strongly disagree" to "Strongly agree". You can also skip a question by selecting "No answer". As a last question, there is also possibility to give freeform feedback. Totally this will take about five minutes only. Survey is open until 13.3.2019 (EOD).

This survey is executed in Webropol, which is one of the biggest online survey companies in Nordic. Survey is done fully anonymously, meaning that only your e-mail address has been added to system to send you a personal invitation link, but that personal information will not be available in any reports.

Thank you for your input!

Best Regards
Tero Lappalainen

1. Product Development Process is clear and easy to follow. *

Think about the Product Development Process end-to-end from raising a request to having new part used in event from your own role point of view.

○ Strongly disagree

○ Disagree

○ Neutral

○ Agree

○ Strongly agree

○ No answer

2. Product Development Process supports design and production of quality components. *

Think about how usual it is to have design, manufacturing or testing of requested parts redone.

○ Strongly disagree

○ Disagree

○ Neutral

○ Agree

○ Strongly agree

○ No answer

3. Request prioritisation in Product Development Process is clear and helps to focus to most important requests. *

Think about whether you need to assess or validate which of your assignments should be done next.

○ Strongly disagree

○ Disagree

○ Neutral

○ Agree

○ Strongly agree

○ No answer

4. Product Development Process provides feedback about request status and results. *

Do you get enough information to readily start working on new requests? Are you able to get feedback how well requested parts performed?

○ Strongly disagree

○ Disagree

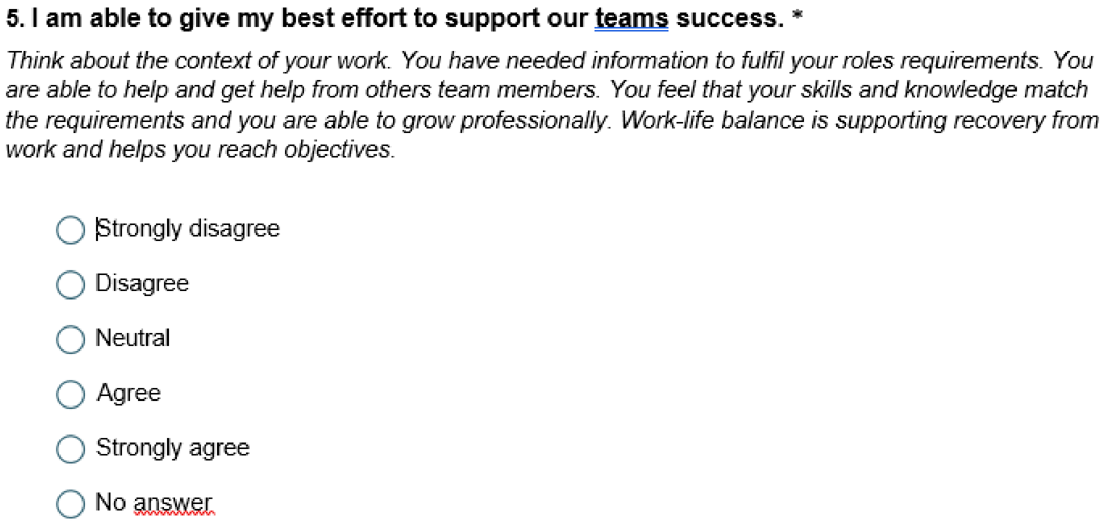
○ Neutral

○ Agree

○ Strongly agree

○ No answer

Appendix 9/3



5. I am able to give my best effort to support our teams success. *

Think about the context of your work. You have needed information to fulfil your roles requirements. You are able to help and get help from others team members. You feel that your skills and knowledge match the requirements and you are able to grow professionally. Work-life balance is supporting recovery from work and helps you reach objectives.

○ Strongly disagree

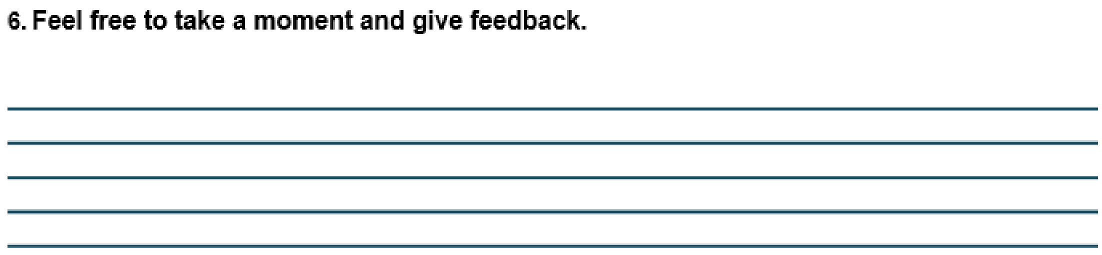
○ Disagree

○ Neutral

○ Agree

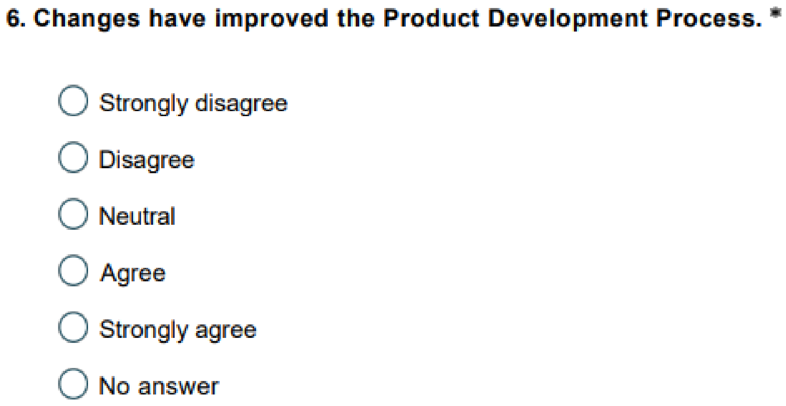
○ Strongly agree

○ No answer



6. Feel free to take a moment and give feedback.

On the second round of survey there was an extra question before free-form feedback:



6. Changes have improved the Product Development Process. *

○ Strongly disagree

○ Disagree

○ Neutral

○ Agree

○ Strongly agree

○ No answer

Appendix 10

WEB-SURVEY RESULTS

Statements:

1 Product Development Process is clear and easy to follow.

2 Product Development Process supports design and production of quality components.

3 Request prioritisation in Product Development Process is clear and helps to focus to most important requests.

4 Product Development Process provides feedback about request status and results.

5 I am able to give my best effort to support our team’s success.

6 Changes have improved the Product Development Process.

Response options:

1	Strongly disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly agree
<empty>	No answer

Before survey	S 1	S 2	S 3	S 4	S 5
Response 1	2	2	1	1	2
Response 2	1	1	1	1	2
Response 3	2	2	3	3	4
Response 4	2	3	1	2	3
Response 5	2	3		4	2
Response 6	4	4	1	3	4
Response 7	2	3	4	1	4
Response 8	3	4	3	2	5
Response 9	3	3	3	3	1
Response 10	2	2	2	1	4
Response 11	3	4	4	3	5
Response 12	2	3	2	2	3

After survey	S 1	S 2	S 3	S 4	S 5	S 6
Response 1	3	4	4	3	4	3
Response 2	4	2	2	3	4	4
Response 3	5	4	4	3	4	4
Response 4	4	4	2	4	4	4
Response 5	3	3	4	4	4	5
Response 6	4	4	3	3	4	5
Response 7	5	4	4	5	3	4
Response 8	4	4	5	4	3	5
Response 9	4	3	4	3	4	5
Response 10	5	5	4	4	5	4
Response 11	5	4	5	3	4	5
Response 12	4	3	4	3	4	4