



CHALMERS
UNIVERSITY OF TECHNOLOGY



Agile Methods in Hardware and Complex Systems Development

An Integration of an Agile Framework in a Traditional Phase-Gate Product Development Methodology

Master of Science Thesis in Master Degree Programme, Product Development

ADRIAN DE ROOS

MASTER'S THESIS

Agile Methods in Hardware and Complex Systems Development

An Integration of an Agile Framework in a Traditional Phase-Gate Product Development

Methodology

ADRIAN DE ROOS

SUPERVISOR & EXAMINER

DR. GÖRAN GUSTAFSSON

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Agile Methods in Hardware and Complex Systems Development

An Integration of an Agile Framework in a Traditional Phase-Gate Product Development Methodology

© Adrian de Roos

Supervisor & Examiner: Göran Gustafsson, Ph.D.

Chalmers University of Technology
Department of Industrial and Materials Science
SE-412 96 Gothenburg
Sweden
Telephone: +46(0)31-772 1000

Gothenburg, Sweden, 2018

Abstract

The purpose of this research was to find out how well an Agile Product Development Framework can be integrated with a traditional Phase-Gate Product Development Methodology.

With the last couple decades of proven successful use of Agile Methodologies in software development, the question has multiple times been risen how well the methodology can be used in hardware development. The study involved investigation of one of the most common used Agile Frameworks, namely Scrum, to gain knowledge about how it is conducted in software in order to ideate and adapt it to hardware development. In addition, a company that specialises on Scrum / Agile in both software and hardware was another source of knowledge, providing with insight through interviews and observations of pilot teams that utilised their methodology.

The Framework was established with lots of resemblance to Scrum but with slight deviation and adaptation to fit the specific conditions of the hardware development pursued at the company that the study was conducted at. The company had a well-established Phase-Gate Product Development Methodology that the new Agile Methodology integrated with to create a Hybrid Framework of the two.

The result consisted of mostly positive outcome with short cycle times and high team satisfaction. The company had the ambition to launch additional pilot projects that utilise the Hybrid Framework.

In conclusion, an Agile Framework can be integrated to a traditional Phase-Gate Product Development project but needs adaptations to the specific characteristics of the project.

Key words: Product Development, Agile, Phase-Gate, Process, Innovation, Hardware, Scrum, Engineering, Lean.

Acknowledgements

I would like to thank Dr. Peter Schubert, Global Process Manager and supervisor at Hilti, for the continuous support and mentorship throughout the internship. The work conducted during the internship provided me with material that set the foundation for the research. Additionally, I want to thank Dr. Luca De Benedetto, Head of Global Process Management, Corporate Research and Technology at Hilti for valuable insights and discussions about the topic and including me in the operational efforts at the department of Product Portfolio Management. I want to show special gratitude towards the team I was a part of throughout the research and all the members of the Pilot Projects that let me observe and document their work and interview them. I would like to thank Hilti for providing me with the incredible opportunity and unforgettable experience to write my thesis and pursue my internship at the headquarters in Schaan, Liechtenstein. In addition, I would like to thank all the other interns I met along the way that made the experience at Hilti even better. Furthermore, I want to thank MBA Klaus Madlener and M.Sc. Alois Jäger from ALEAS for letting me interview and give me continuous valuable information for the research and methodology.

A special thanks to Dr. Göran Gustafsson, Senior Lecturer at the division of Product Development, Department of Industrial and Materials Science and my thesis supervisor and examiner at Chalmers University of Technology for always showing a genuine interest and enthusiasm and providing with invaluable answers and feedback to my inquiries and progress of the research. I am grateful that Göran gave me lots of inspiration in the topic of Product Development and related areas.

Last but not least I would like to thank Chalmers University of Technology for providing me with an excellent education from day one until my graduation.

Table of contents

1	Introduction	1
1.1	Background.....	1
1.2	Aim	2
1.3	Delimitations.....	2
1.4	Outline of the Thesis	2
2	Problem Formulation	3
2.1	Problem Definition	3
2.2	Research Question and Expected Outcome.....	3
3	Methodology.....	4
4	Theoretical Framework.....	5
4.1	The 5 W's and 1 H	5
4.2	Waterfall	5
4.3	Phase-Gate.....	6
4.4	Lean Product Development.....	7
4.5	Agile Project Management.....	9
4.5.1	Agile Manifesto.....	9
4.5.2	Scrum	10
4.5.3	Scrum-of-Scrums / Meta Scrum.....	16
4.5.4	Kanban.....	16
4.5.5	Comparison of Scrum and Kanban.....	18
4.6	Hilti V-model / Requirements Engineering.....	19
4.7	The Hilti Technology Process (TP)	21
4.8	The Hilti Time To Money (TTM) process	22
5	The Hilti Agile Hybrid.....	24
5.1	Pilot Projects	25
5.1.1	Pilot project A	25
5.1.2	Pilot Project B	25
5.1.3	Pilot Project C	25
5.2	The Hilti Agile Guideline / Documentation	26
5.2.1	The Sprint.....	26
5.2.2	Agile Materials and Tools	26
5.2.3	Roles and Responsibilities.....	41
5.2.4	Meeting Culture	43
5.2.5	Preparation for Project Execution	53
6	Discussion, Conclusions and Recommendations	56

6.1	Discussion and Reflections.....	56
6.2	Research Aims	59
6.3	Challenges and Problem Formulation Revisited.....	59
6.4	Research Question	60
6.5	Future Work.....	60
7	References.....	61

List of Abbreviations

HW = Hardware

SW = Software

OTD = On Time Delivery

5W1H = 5 Why's 1 How

SM = Scrum Master

SMART = Specific, Measurable, Actionable, Realistic and Time-bound

PO = Product Owner

POT = Product Owner Team

ROI = Return On Investment

VoC = Voice of the Customer

DoD = Definition of Done

WiP = Work in Progress

RUP = Rational Unified Process

RE = Requirements Engineering

REQ = Stakeholder Requirements

RT = Requirements Testing

SPEC = System Specification

ST = System Test

DES = System Design

DT = Design Specification Testing

IMP = Implementation

CCT = Customer Confidence Test

TP = Technology Process

TTM = Time To Money

G = Gate

RTB = RealTime Board

TTM PL = Time To Money Project Leader

TPL = Technology Process Leader

BU PM = Business Unit Project Manager

RIV = Risk Impact Value

SP = Sprint

CAD = Computer Aided Design

Obeya = Japanese for project meeting room or “Big Room”

1 INTRODUCTION

With fierce competition in Hardware (HW) Development, where constant product innovation is essential to stay ahead of other market actors and maintaining market shares, heavy emphasis is put on reducing innovation cycles while keeping superior quality. In order to do so, new innovative methodologies are necessary. By taking best practises, conducting internal research and gaining inspiration from adjacent areas, a company can take a leap forward and increase their chances of gaining market shares.

One of the areas that has been transformed in the way of managing projects is Software (SW) development by using a Framework named Agile Project Management. The Agile Framework's foundation is based on the Agile Manifesto which was written in 2001 by seventeen independent-minded and non-agreeing software practitioners. These practitioners had four main values that they shared and so was the manifesto created (Agile Alliance, 2001). The Agile Software Development Framework bases its way of working on continuous improvement and with cross-functional teams taking constant small steps when developing the product. There are several ways of working with Agile Software Development but the most common one is Scrum (Testingexcellence, 2017).

Since the Agile methodology has had its most prominent success in Software Development, some argue that the methodology can be applied to HW Development as well. It can be seen that Agile principles were used in improvements in Japanese and American manufacturing companies as early as 1976 (Takeuchi & Nonaka, 1986). However, there are challenges for different reasons to implement Agile methodologies in HW development because it might have to be adapted to fit the specific way of working in the organisation.

1.1 Background

Hilti is a privately owned multinational company that provides leading-edge tools, technologies, software and services in the construction industry. Hilti is working closely with its customers, run their own research and design labs and collaborate with top universities around the world. The company was founded in 1941 in Schaan, Liechtenstein where it still has its headquarter. Today the company has over 25 000 employees and is present in more than 120 countries (Hilti, 2018). What distinguishes Hilti from many of their competitors is their big sales force, making the company have extensive direct contact with its customers, thus cutting off the middle-man (e.g. retailer).

Hilti has the intention to further distinguish themselves as world-class leader in the industry and to do so, Hilti aims to reduce cycle time in innovation by implementing new methodologies in Hardware Development (HW) as a part of the revolution in their global process Framework. According to Hilti themselves, they are currently best-in-class in applying Phase-Gate development Framework and by integrating new process methodologies with their present, a significant improvement is expected.

1.2 Aim

The strategic goal is to improve the overall productivity in Hilti's innovation projects. Through Hilti's evolution in their Global Process Framework, Agile/Scrum will be a contributing methodology.

Results expected include:

- A documented concept utilising Agile/Scrum in hardware and complex systems development which integrates with current Phase-Gate methods
- A contribution to the definition of the business unit

1.3 Delimitations

Investigation of the effect from changes will primarily be gathered from the pilot projects that are incrementally trying out adaptations to find the best methodologies.

Focus will not be applied to software development from which Agile/Scrum originally derives but instead on HW development and complex systems, e.g. large mechatronics. The thesis research lasts 5.5 months.

1.4 Outline of the Thesis

This chapter intends to provide an overview of the general structure of the thesis and what the various components are.

Chapter 2 explains the thesis main topic and what challenge that the company faces that they want assistance with and what outcome that is expected from the research.

Chapter 3 covers how the author approached the topic, which methodologies and tools that were used in addition to where the author was situated to consolidate the information needed. Furthermore, the stakeholders of the thesis are stated.

Chapter 4 covers the theoretical Framework, i.e. the literature review and all the information that lays the foundation for the research to be comprehended.

Chapter 5 contains the outcome of the research and material provided to the company made by the author including reflections.

Chapter 6 concludes the thesis with discussions, recommendations and an assessment of how well the research aims were reached. This chapter also includes a little bit about what can be done in the future with the thesis topic.

2 PROBLEM FORMULATION

This chapter provides a description of the problem and from there a definition of the Research Question that the study intends to address paired with an expected outcome.

2.1 Problem Definition

Hilti wants to shorten the cycle time in the innovation processes in HW development. Since 2017, there have been several HW development pilot projects which incorporate various Agile methodologies in the well-established Phase-Gate product development process, namely Time To Money (TTM) and Technology Project (TP).

Because the Agile Framework, especially Scrum, is most extensively used in SW and thus best adapted to SW development, some inherited challenges have been identified by Hilti. The Agile Framework, especially Scrum, might not be able to be implemented in its entirety, there will have to be adaptations to fit Hilti the best way. Some of the challenges are:

- Interdependencies between internal (activities in engineering and manufacturing) and external (suppliers, integration, approvals, etc.)
- Cross-functional and cross-organisational project contribution
- When global Project Teams are not co-located
- Making the final stakeholder a part of the team

In addition, since the Phase-Gate in TTM and TP processes has been the main development methodology in HW development for a long time, there is a potential that some of the employees are resistant to changing their way of working, making it harder to implement an Agile Framework. To make a roll-out of the methodology throughout the organisation, a sustainable solution of the Hilti Agile Hybrid with high commitment from the practitioners is necessary to ease the process of rolling out.

With the implementation of Agile Framework, it is expected to get higher transparency, clearer project goals, continuous improvement, ability to react to changes and integrate customer feedback and have a better cross-functional collaboration.

2.2 Research Question and Expected Outcome

The focus of the research question is to create specific concepts of how to integrate the agile mind-set and methodologies with already existing methodologies. In order to substance results, pilot projects will be consulted and measured which the author will be in continuous contact with. The specific topic that this thesis will investigate is:

RQ	<i>How can Hilti adapt and integrate an Agile Methodology with their Phase-Gate Methodology in Hardware Development?</i>
Expected Outcome	<i>A comprehensive documentation with step-wise instructions, applicable and adapted to fit multiple categories of Hilti HW Development</i>

The documentation includes the author's reflections and analyses that questions and/or justifies the methodology and approach.

3 METHODOLOGY

This section intends to give the reader an overview of how the research was conducted and which methods that were used in the study.

As a starting point, facts regarding the definition of Agile/Scrum and relevant theories was researched, where heavy emphasis was put into literature review regarding the foundation of the methodologies and more specifically what adaptations that had to be made in order to deploy the methods in hardware development. Knowledge deriving from various courses at Chalmers University of Technology was utilised. Furthermore, relevant subjects touching these methodologies, such as Lean Product Development was briefly investigated.

Tools that was extensively used in acquiring relevant information were observations of meetings and interviews with people from various Pilot Projects that have been deployed at Hilti. In addition, consultation of experts from different Business Units (BUs) at Hilti were integrated in the approach.

Since management and development methodologies are in constant change, the author made sure to get input from different sources and people with relevant experience in state-of-the-art Agile/Scrum methodologies in order to create a personal perception of what would suit Hilti's specific situation best. This was in continuous consultation with stakeholders such as the supervisor, superiors and experts.

The author belonged to the department of Product Portfolio Management at Hilti which focused on aligning processes, methods and tools. Since a big part of the thesis included constant improvement of the method developed, the author was physically situated at the company's headquarters in Schaan, Liechtenstein. Considering the task involved consolidating Hilti's present methodologies in hardware development with best practises from pilots together with the author's own research, a close communication with the various BUs was established.

To gain hands-on knowledge and make thorough empirical research to assist the company with the topic, the author created a comprehensive documentation and training material for newly established projects that sought to use the Hilti Agile Hybrid. Much of the information was gathered from observation of teams that are executing product development (piloting) and adapting an Agile Framework and with that information extract best practices.

The main stakeholders of the project are:

- Supervisor and examiner at Chalmers University of Technology (Dr. Göran Gustafsson)
- Vice Head of Department at Industrial and Materials Science, Chalmers (Dr. Lars Almefelt)
- Director of Master's Programme Product Development, Chalmers (Dr. Erik Hulthén)
- Superiors at Hilti (Dr. Peter Schubert, Dr. Luca De Benedetto)
Various business units at Hilti, especially Product Portfolio Management
- Experts in Phase-Gate and current development methodologies at Hilti

4 THEORETICAL FRAMEWORK

This chapter outlines the theory that is necessary to be familiar with when reading this thesis. It serves as a support in understanding of the content and where some of information in the research derives from.

4.1 The 5 W's and 1 H

The 5 W's and 1 H are six questions that can be asked to get a deeper level of knowledge in a structured way resulting in a complete story on a subject. The method is often used in for example police investigations and journalism (Don, 2016). The questions are:

Why	▶	Why is the task / event etc. executed?
What	▶	What do we need to do?
Who	▶	Who are the stakeholders, team, customers, sponsor etc.?
When	▶	When will it take place?
Where	▶	Where will it be delivered, or take place?
How	▶	How is it done? Methods, practices, tools etc.

4.2 Waterfall

The waterfall methodology is an approach for development of products in a linear fashion and is widely used in engineering where it was initially developed in the manufacturing and construction industries. The work is split into phases or steps and the flow is defined as logical steps where one must be fulfilled before you can start the succeeding phase (Wikipedia, 2018).

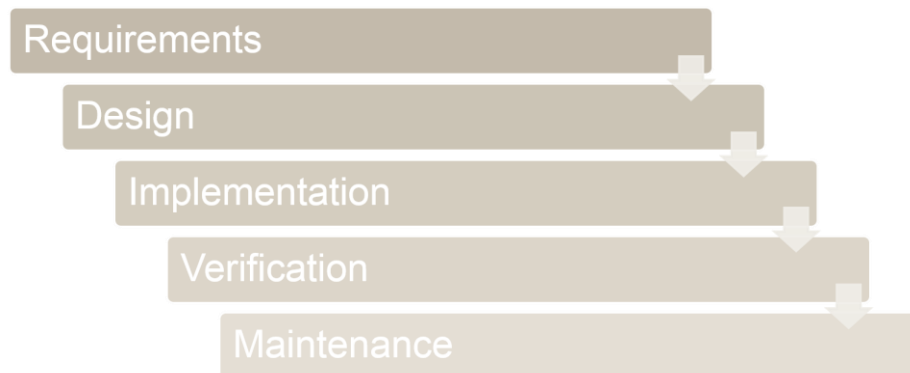


Figure 1. The waterfall methodology

The Waterfall methodology includes the following steps: **Requirements Determination, Design, Implementation, Verification** and **Maintenance** where all the steps are carried out sequentially, as can be seen in Figure 1. The methodology makes the assumption that all the requirements can be found early in the development project where most planning for the project is performed, therefore a lot of communication with the user is necessary in this phase (UMSL, 2009). This might be seen as an overoptimistic statement because the chance of requirements changing in a long-term project is high. The process proceeds with Design in which the system architecture and the necessary hardware and software is defined. The phase can be split into two parts: the logical and physical design where in the logical phase, analysts uses the information from the Requirements phase to design the system independently between hardware and software. Thereafter, the analysts transform the hardware and software requirements into physical design (UMSL, 2009). When all the code is written, the

Implementation phase can start which is then followed by Verification, which is the phase where the testing of the product takes place to see if it meets the customer expectations. After Verification, the user will be using the product and thus the Maintenance phase takes place where found errors are being corrected (UMSL, 2009).

4.3 Phase-Gate

The Phase-Gate is a methodology that has a waterfall mind-set with a set of activities, analyses and deliverables between every gate. These activities generate results which are analysed and then converted into deliverables and depending on how the deliverables stack up to the pre-defined expectation, a decision as to whether or not that project will continue (Go) or be aborted (Kill) is made (Cooper, 2008).

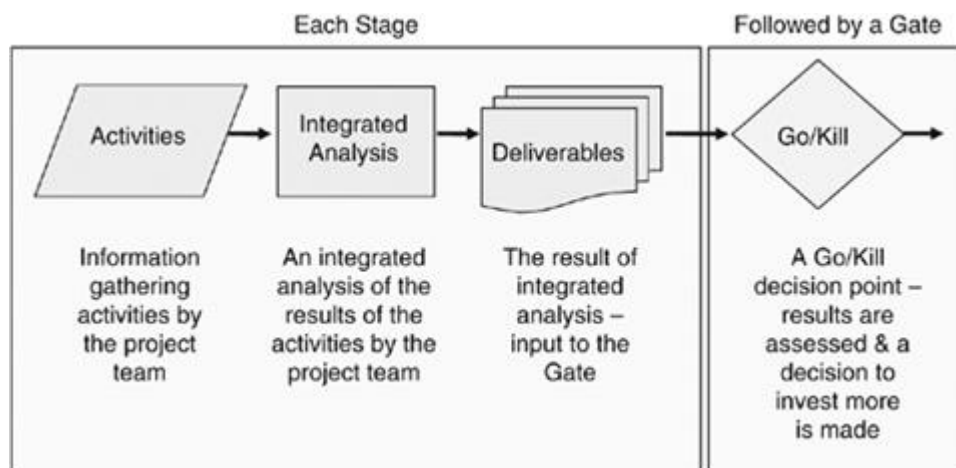


Figure 2. Stage and gate in the Stage-Gate® process (Cooper, 2008)

The Stage-Gate® process is one of the most common Phase-Gate methodologies and consists of a series of stages that starts with *discovery* and ends with *post-launch review* (Cooper, 2008) as can be seen in Figure 2. A typical Stage-Gate® process for large projects can look like the one depicted in Figure 3. Within the stages, the tasks are worked on concurrently and the gates serve as a quality control.

Very early in the process, “Phase 0”, brainstorming is usually taking place in order to find an idea that the company wants to pursue. The ideas are scanned and different stakeholders and customers are consulted to find out how and why they use products before the process enters the Scoping stage, Stage 1. During the scoping phase the project makes a scan for technical feasibilities and conducts a market research. It is important to know who the customers are and what market the product is in, as well as getting familiar with the competition. The results from the scoping stage are used to make a proper business case, Stage 2, which is essential to assess before developing the product. This stage is usually divided into four cornerstones: product definition and analysis, the business case, the project plan and the feasibility review. The business case justifies why the company wants to make the product in the first place, the project plan is a list of tasks and activities throughout the product development including milestones, needed financial resources for each task and launch date among several. The feasibility phase ensures that the previous information that has been gathered guarantees that the right product will be developed (Anon., 2018). When entering Development, Stage 4, the plans that were made in previous stages are executed with testing and the product’s marketing and production plans. A time plan with milestones is defined that follows the SMART (specific, measurable, actionable, realistic and time-bound) guideline which is constantly

being reviewed and updated to make sure that the team stays on task and informs the management about the progress of the product. Entering Stage 4, the Product Launch stage, emphasis is put on finding and fixing bugs and issues. It should be noted that at this stage, the product is no longer just a prototype but a nearly finished product with almost all the features and designs that the final commercialised product will have. This is a major difference between the Phase-Gate and Lean Product Development (detailed in the next chapter) where the project, being at this stage, already has a minimum number of errors because these have already been mitigated through constant reviews in earlier phases. The stage includes field testing to gain valuable feedback on the product and inspect how the customer actually uses the product and in which environments. This information is helpful for the marketing team to know how to focus their efforts in sales. Stage 5 is the final step which can only be reached if all previous gates have been passed. This stage consists of providing sales and support personnel to get familiar with the product and a strategy to get customer attention for the product (Anon., 2018). Finally, when the product is launched, a debrief of what went well and what could be improved is conducted which can contribute to future projects' success (Edgett, 2018).

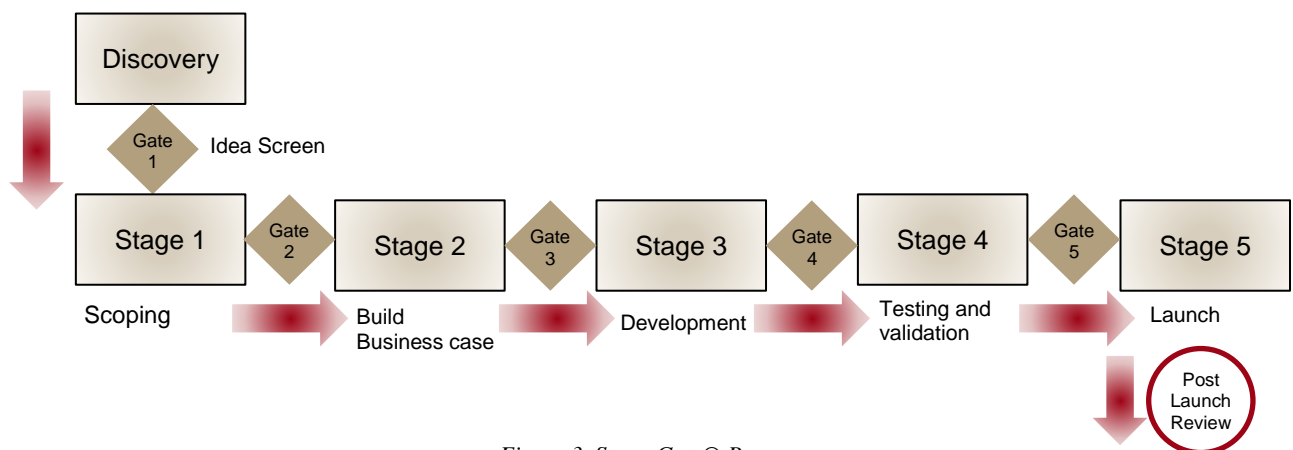


Figure 3. Stage-Gate® Process

4.4 Lean Product Development

The word “Lean” originates from Toyota Product System methodology that was developed to spot and eliminate waste. This Lean manufacturing process resulted in a number of principles that the manufacturing lines adapted (Mynott, 2012). The phrasing expanded to the actual development phase, also known as Lean Product Development (LPD), where the intention was, just like in manufacturing, to eliminate waste. The issue is that most of the waste is not very apparent because most of it is either on hard-drives or in engineer’s brains (Mynott, 2012). Companies often struggle with not developing the right product that the customer actually wants, therefore, lots of the focus in Lean PD is on continuous improvement.

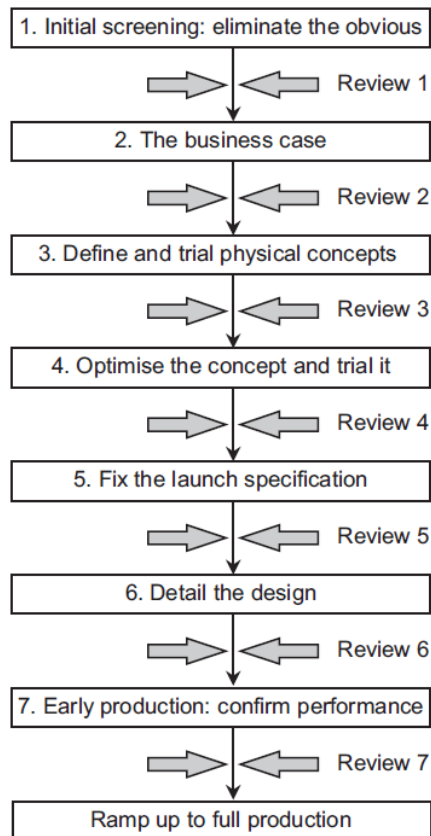


Figure 4. The phases and Reviews of Lean PD (Mynott, 2012)

Lean PD has, in essence, 7 phases and an equal number of reviews that work, just like in Phase-Gate, as a Go/Kill for the project. The phases are described briefly below (Mynott, 2012):

The first phase makes sure that the project is in line with the company strategy.

The second phase includes finding the customer requirements, technical feasibility and cost estimation.

Phase three covers model generation with trials of unproven elements, some of which might not be good for this particular project but may instead be kept for future efforts.

Phase four takes the best features and combine them into an optimal product which is then tested including a draft launch preparation.

Phase five details the launch- and production process specifications.

Phase six includes providing the plant with correct drawings to eventually initiate the production.

Phase seven covers early production and trials of production to confirm the previous specified costs.

When all the relevant gates have been passed, it is possible to start the ramp-up of production

Lean PD differs from the traditional Phase-Gate especially in the amount of testing and trials that are being conducted in order to ensure that the project does not get invested in too much without knowing if the system is actually feasible / operational. Failing fast does not let the project not become too costly and sensitive to changes at later stages, it is thus preferred to identify issues as early as possible in product development.

4.5 Agile Project Management

One of the most central Frameworks for the topic of the thesis is the Agile Project Management Framework which is normally heavily used in software. This section explains the fundamentals of the Framework.

4.5.1 Agile Manifesto

The foundation of the Agile Project Management Framework builds on the four pillars that were developed in 2001 (Agile Alliance, 2001).

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

That is, while there is value in the items on the right, we value the items on the left more.” (Beck, et al., 2001)

What *Individuals and interactions over processes and tools* means is that the agile technique will facilitate and put more focus on the people and the communication instead and then the essential processes and tools will be decided afterwards. *Working software over comprehensive documentation* implies that it is hard to know all the requirements of the customer from the beginning as they will most likely change during the process of developing the product or that the customer herself might not even know what she wants. Instead, with the Agile approach, incremental shippable sub-products and continuous contact with the customer that can see the actual product finalising is established. Therefore, comprehensive documentation is not as necessary like in traditional approaches. *Customer collaboration over contract negotiation* has its foundation in that a very rigid contract already from the beginning is a false security in a way that you assume that you know everything from start. The rigid contract makes it harder to include changes and this is where the iterative methodology is strong, so, with an Agile approach contracts are still signed but they need to be flexible to let the customer be in control. *Responding to change over following a plan* implies that instead of having a comprehensive and detailed plan throughout the project that is virtually impossible to keep, the project plans are exchanged with release schedules and “Burndown” charts (charts that indicate how much of the project or sub-project is finished at a certain point of time) which facilitates changes (Ocamb, 2013).

To complement the four pillars, twelve principles were also formulated. These clarify what it means to be agile, the intent and the culture that comes with the Framework. The principles are (Agile Alliance, 2001):

1. **Customer satisfaction through early and continuous software delivery** – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
2. **Accommodate changing requirements throughout the development process** – The ability to avoid delays when a requirement or feature request changes.
3. **Frequent delivery of working software** – Scrum accommodates this principle since the team operates in software Sprints or iterations that ensure regular delivery of working software.
4. **Collaboration between the business stakeholders and developers throughout the project** – Better decisions are made when the business and technical teams are aligned.
5. **Support, trust, and motivate the people involved** – Motivated teams are more likely to deliver their best work than unhappy teams.
6. **Enable face-to-face interactions** – Communication is more successful when Project Teams are co-located.
7. **Working software is the primary measure of progress** – Delivering functional software to the customer is the ultimate factor that measures progress.
8. **Agile processes to support a consistent development pace** – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
9. **Attention to technical detail and design enhances agility** – The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.
10. **Simplicity** – Develop just enough to get the job done for right now.
11. **Self-organising teams encourage great architectures, requirements, and design** - Skilled and motivated team members who have decision making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
12. **Regular reflections on how to become more effective** – Self-improvement, process improvement, advancing skills, and techniques help team members' work more efficiently.

In conclusion, the Agile Framework is customer focused and encourage customer guidance and participation.

4.5.2 Scrum

Scrum is a lightweight framework that facilitates the development of complex products. The team consists of a number of people working together in short, sustainable bursts of activity called Sprints including time to reflect and review of the result from the Sprint (Sims & Johnson, 2012). Scrum is an Agile Framework with iterative product development, making it

easier to create changes throughout the project. The process is illustrated in Figure 5 including the artefacts and events which are described in further detail later.

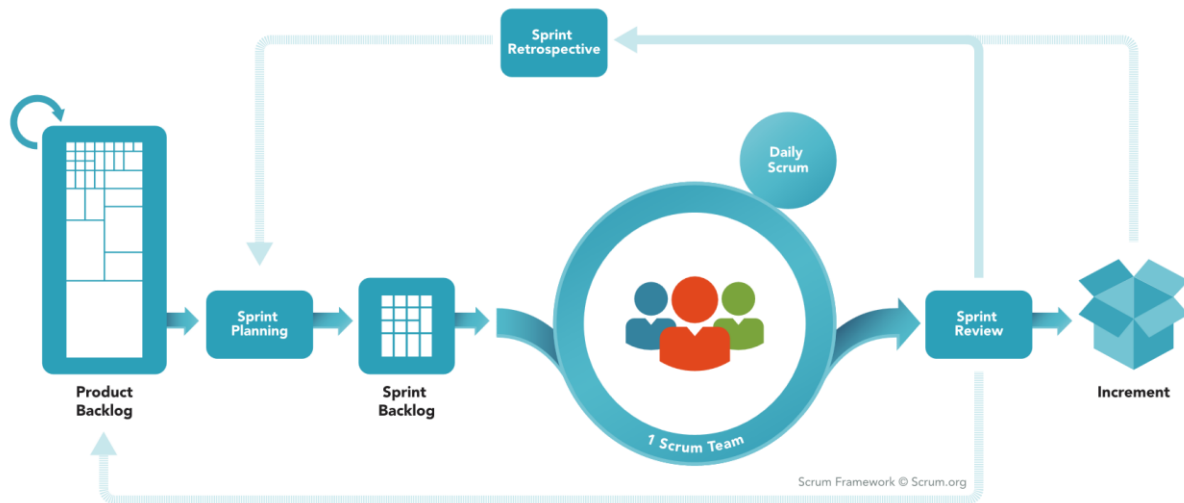


Figure 5. The Scrum Framework (Scrum.org, 2016)

Scrum roles and responsibilities

The Scrum Framework includes three roles: Scrum Master (SM), Project Owner (PO) and Project Team and they all carry distinct duties and responsibilities. Apart from the three appointed roles, other stakeholders are identified as well, such as:

- The customer, who is normally the financier of the project
- The user, who provides the project with valuable requirements inputs
- The managers, who creates the organisational frame.

Product Owner (PO)

The PO has his/her primary role of being financially responsible for the product and to optimise Return on Investment (ROI) by always being in contact with the stakeholders to find out what exactly they want to have developed in order to make the right product, i.e. representing the Voice of the Customer (VoC). The following are the responsibilities (Maximini, 2015):

- Create a product vision that describes the purpose of developing the product
- Identify and refine requirements to formulate them well for the Product Backlog Items in form of User Stories that contain a clear and understandable business value
- Manage and prioritise the Product Backlog (see Artefacts) and be accountable for that
- Product strategy development
- Feasibility studies
- Help the team to understand *what* to develop and *why*
- Review and accept the delivered product increment
- Authority over the requirements, not over the estimates or the people

- Inform stakeholders regarding progress and expectations

Scrum Master (SM)

The SM is the facilitator and supports the PO and the teams' work towards their Targets and protects the team from external interruption. It is important that the SM is experienced with Scrum as he/she will be managing the process of the product development. The SM's responsibilities are (Maximini, 2015):

- Increasing productivity of the Scrum Team by removing impediments
- Making sure the Scrum process is understood and adhered to
- Making problems (called "impediments") transparent and solving them
- Training project participants in Scrum
- Focusing project participants on the project goals
- Facilitating events
- Communicating with anyone interfacing with Scrum

It is important that the SM has a good amount of soft skills as he/she does not have responsibility for any specific technical domain. Apart from having an open, motivating, disciplined and courageous character, some of the soft skills needed are (Maximini, 2015):

- Communication skills
- Conflict resolution skills
- Moderation/facilitation skills
- Time management skills
- Negotiating skills.

Project Team

The Scrum Team is a self-organising group of people from the different necessary functions who develop the product and are responsible for delivering the results from each Sprint. The team works closely with the PO and some of their responsibilities are:

- Support in refining the Product Backlog items so that they are easily understood
- Together with the PO, create the Targets and the estimates that are achievable during the Sprint
- Track their own progress
- Depending on level of trust, decide who they want on their team (Maximini, 2015).

Scrum Artefacts

The mandatory artefacts included in Scrum are: The **Product Backlog**, the **Sprint Backlog** and the **Product Increment**. All the artefacts contribute by: reducing the amount of documentation and optimising the process through regulatory inspections and adaptations.

The artefacts are transparent in the sense that they are always accessible. In addition to the mentioned three artefacts, a number of voluntary artefacts are also included.

Product Backlog

The product Backlog has all the requirements for the product which the Project Team will deliver throughout the project and it is the PO who is responsible for the product Backlog. In collaboration with the Project Team, the product Backlog items are made sure to be well formulated and understood in addition to ensuring that the right items are implemented. The items that will be worked on during the succeeding one or two Sprints are described in more detail in comparison to items that will be worked on later in the project (Maximini, 2015).

Sprint Backlog

The Sprint Backlog includes items that come from the product Backlog and will be achieved by the Project Team during the current Sprint. In the Sprint Backlog, the breakdown of how the tasks will be turned into a finished product increment is also stated. The Sprint Backlog is made by the Project Team during the Sprint planning and can be changed, refined and updated throughout the Sprint (Maximini, 2015).

Definition of Done

Definition of Done (DoD) is one of the most important tools used by Scrum teams to ensure quality. When DoD is reached, the product Backlog item is considered finished. In essence, the DoD is an agreement between the PO and the Project Team about what measures are necessary to be taken for every product Backlog item. The measures can be for example (Maximini, 2015):

- Complete all tasks with no work remaining
- Verified acceptance criteria through tests
- Update documentation

However, each DoD is customised for the specific demands of the product and it makes sure that everyone understands what is required to say that it is “done” (Maximini, 2015).

Scrum Values

The most central and extended values of Scrum are depicted in Figure 6:

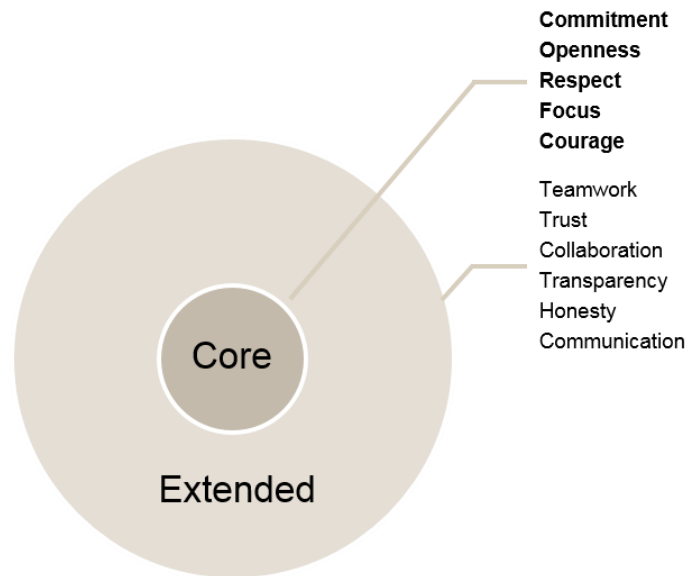


Figure 6. Core and extended Scrum values (Maximini, 2015)

The Scrum Alliance describes the core values in the following way (Scrum Alliance, 2001):

- **Focus:** Because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner.
- **Courage:** Because we work as a team, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges.
- **Openness:** As we work together, we express how we're doing, what is in our way, and our concerns so they can be addressed.
- **Commitment:** Because we have great control over our own destiny, we are more committed to success.
- **Respect:** As we work together, sharing successes and failures, we come to respect each other and to help each other become worthy of respect.

Scrum Events in the Framework

This chapter gives an overview of the different events that occur for teams that utilise the Scrum Framework

Sprint

The Sprint is the heart of Scrum which contains the Sprint Planning, Daily Scrums, the actual development work, the Sprint Review and the Sprint Retrospective (Scrumguides, 2017).

The Sprint is constrained to one month or less time and a new Sprint always starts with the end of the previous one. The intention with the Sprint is to reach the Sprint Goal which in turn is a deliverable (Scrumguides, 2017).

Sprint Planning

The Sprint Planning meeting is split into two parts. The first part includes deciding what the team should achieve and deliver after the Sprint. The PO presents a set of user stories in priority order which the team discusses together with the PO and reviews the acceptance criteria to get a good understanding of what exactly is necessary in order to reach the expected outcome from the user story. After this it is up to the team to decide how much time is needed for the work to be done. This is repeated for as many user stories that the team thinks that they can manage during that Sprint. The second part includes how the team intends to perform the work by decomposing the user stories into tasks such as “get input from users” or “run release scripts” (Sims & Johnson, 2012).

Planning Poker

To estimate the effort that is needed to complete a user story, the team can utilise what is called “Planning Poker”. The origin comes from software development where expert judgment-based estimates had a tendency to be over-optimistic, especially when estimating their own work (Mahnič & Hovelja, 2012). The estimation technique is widely used in Agile projects, especially Scrum. The user stories that the Product Owner brings to the team are jointly given estimates by the Project Team in terms of a pre-defined set of values, e.g. 0.5, 1, 2, 3, 5, 8, 13, 20, 40 and 100. *“Participation of people with different backgrounds helps in reducing the over-optimism of expert judgment-based estimates by identifying more issues affecting implementation”* (Mahnič & Hovelja, 2012). The technique starts with the Product Owner explaining the requirements and then the team discusses what work that needs to be done to achieve the user story. Each team member then writes his/her estimation (a value from the set number sequence) on a card. These cards are then revealed at the same time by all members followed by the highest and lowest number explaining the reasons for their choice of value. The process is then repeated until consensus is achieved.

Sprint Review

The Sprint review indicates the end of the Sprint. This meeting is intended for the team to show their accomplishments, including what they did not accomplish, through the Sprint for the rest of the team. All the stakeholders are supposed to be present in order to inspect and adapt the increment and give feedback (Sims & Johnson, 2012).

Sprint Retrospective

The Retrospective meeting is important for gathering of learnings and making sure that the team is ever-improving. These learnings will help the team to focus on what they will commit to during the next Sprint. The event often lasts about an hour per week of development, meaning that if the Sprint lasts four weeks, the event is recommended to last approximately four hours (Sims & Johnson, 2012). The Scrum Master focuses on making the coming Sprint more efficient and enjoyable with the help from the previous learnings. By the end of the meeting, the team should have found potential improvements for the next Sprint (Scrum.org, 2016).

Daily Scrum Meeting

The Daily Scrum is sometimes referred to as the “stand-up” meeting because the intention is to make this a very brief summit of maximum 15 minutes and not having feel like it is just another meeting among all other meetings. The meeting is for the team to quickly share information and adapt the team members’ work and surface problems that have arisen. The team does this by letting each participant answer the following questions (Sims & Johnson, 2012):

- *What tasks I have completed since the last daily Scrum*
- *What tasks I expect to complete by the next daily Scrum*
- *What obstacles are slowing me down*

4.5.3 Scrum-of-Scrums / Meta Scrum

When the team is very large, it is possible to divide the team into smaller Scrum teams where each sub-team entitles one member to be the ambassador who in turn participates in daily Scrum meetings with the rest of the sub-team ambassadors. This meeting is known as the Scrum-of-Scrums. The ambassadors can have different roles, such as Scrum Master, technical contributor or manager, in their respective sub-team depending on the context. The meeting structure of the Scrum-of-Scrums is the same as in the regular Scrum - every ambassador represents each sub-team in order to report completions, next steps and potential impediments. The solutions of the impediments are expected to focus on the challenges of coordinating between the teams, these are in turn tracked through a Backlog of its own (Agile Alliance, 2018). The structure of the Scrum-of-Scrums is visualised in Figure 7.

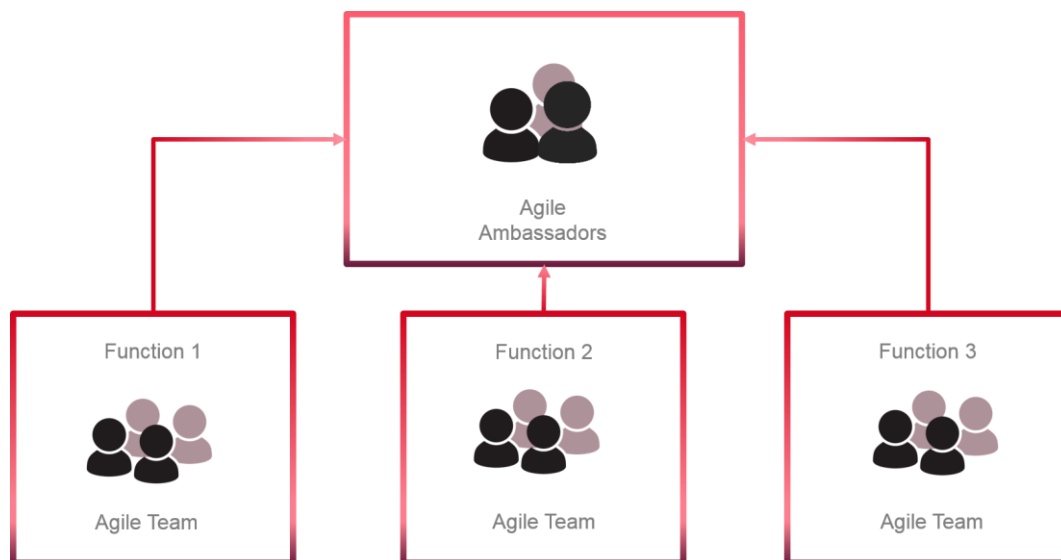


Figure 7. Scrum-of-Scrums

4.5.4 Kanban

“Kanban is based on a very simple idea. Work in Progress (WiP) should be limited and something new should be started only when an existing piece of work is delivered or pulled by a downstream function.” (Kniberg & Skarin, 2010)

Kanban has evolved from SW development but it is flexible enough to be applied to other types of projects. In addition, Kanban can in theory be scaled up and applied to a whole organisation or value chain, but it is mostly used separately in individual parts of the value chain. The Kanban method does not emphasise *how* the product will be developed but *that* it will be developed, meaning more focus on the actual process (Leopold & Kaltenecker, 2015). To successfully implement Kanban there are six core principles to be considered. These are (Leopold & Kaltenecker, 2015):

1. Make work visible.

By applying Kanban, the intention is to establish a continuous workflow that generates more value for the customer. With this it became clear for people exactly with who they should communicate within the organisation to improve their process.

2. Limit Work in Progress (WiP).

Production companies in general want to minimise the number of unfinished projects because they generate tied-up capital. The argument is that it is only when the product is finished and ready to be delivered that the company can charge money for it. It is thus smarter to finalise one operation 100 % instead of having 10 operations at 10 % each so that the company does not have to have a lot of money locked.

3. Manage flow.

Since the Kanban methodology is all about focusing on the workflow, it is necessary to diminish whatever interrupts a smooth workflow like bottlenecks and blockers first before going to new work. Kanban assists with establishing a fast and consistent flow which is predictable.

4. Make progress policies explicit.

With the use of Kanban, the team is committing to a collection of policies which must be transparent to all participants.

5. Implement feedback mechanisms.

Kanban focuses on continuous improvement, and to do so it is necessary to constantly acquire learnings in terms of feedback mechanisms of what could be done better, such as stand-up meetings. It is recommended to have these meetings not just in the team but also on a higher level, i.e. in the value chain. Mechanisms could also include retrospective and Targeted improvement meetings.

6. Improve collaboratively (using methods and models).

Instead of reinventing the wheel, it is more efficient to reuse knowledge and approaches that have been successfully used before. Kanban is an adaptive method for different ways of working for the purpose of development work in general and different organisations.

Make work visible.

Kanban tracks the flow of work by using a whiteboard and sticky notes (or electronic substitute) and provides good transparency which exposes bottlenecks, queues, variability and waste. This visibility also encourages improvements and implementation of these which in turn provides incremental improvements of existing processes that are aligned with Agile and Lean (Kniberg & Skarin, 2010).

On a typical Kanban board, depicted in Figure 8, the different phases are displayed in columns and what is essential on a Kanban board is that every column must have some WiP limit, meaning that a new task cannot be moved from one phase to another unless another task is being finished (Reynisdóttir, 2013).

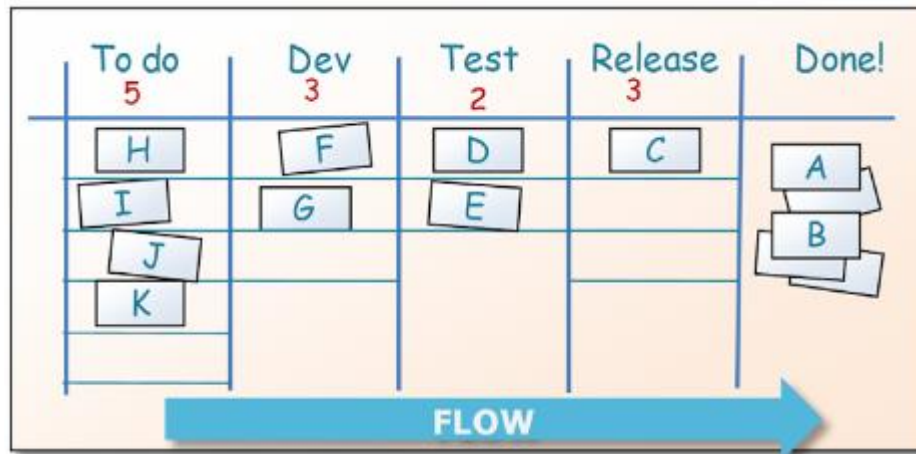


Figure 8. Typical Kanban board (Kniberg & Skarin, 2010)

4.5.5 Comparison of Scrum and Kanban

The information regarding the methods are by the book and in most cases their parameters or rules are changed depending on the organisation/environment.

Scrum and Kanban have many similarities: they are both process tools in the sense that they help you work more effectively. According to Kniberg & Skarin, 2010, Kanban is more adaptive, in contrast to being prescriptive, than Scrum in terms of how many rules they need to follow – three versus nine. But relatively speaking they are both very adaptive compared to some other methods like RUP (Rational Unified Process).

The two differ in the number of tasks that the team can have as WiP. Kanban restricts this number while in theory there is no limit in Scrum. However, many teams which utilise Scrum realise sooner or later that having too many ongoing items is not a good idea and thus try to lower the number of WiP or simply restrict it to a specific number, essentially making it a Kanban board (Kniberg & Skarin, 2010).

Both of the processes are empirical in the sense that they are expected to be used in an experimental manner in order to adjust and customise it according to the users' liking or what suits the organisation / environment. Related, both processes are emphasising continuous improvement where they acquire results, investigate them and draw conclusions of how to improve themselves (Kniberg & Skarin, 2010).

A Scrum team resists change within an iteration. What is committed to by the team from the beginning is what the team will work on for the entire Sprint. If someone wants to add an item, it has to be accepted by the Product Owner which might then put it in the "To-do" or "Backlog" or what the team prefers to call it. This item can then be pulled in to the next Sprint. In Kanban, a change is allowed but only on the condition that the WiP limit is not reached. If it is full, one item has to be removed (Kniberg & Skarin, 2010).

The stickers on a Scrum board move from left to right (from "To-do" to "Done") throughout the Sprint and at the end of the Sprint all stickers are cleared to make space for new ones. The comparison between a Scrum Board process and a Kanban Board process can be seen in Figure 9 and Figure 10. In addition, the Scrum board is owned by a single team and the team

members are cross functional with the right number of skills to finalise the items on the board, whereas in Kanban, the board is not necessarily owned by a single team but more a workflow (Kniberg & Skarin, 2010).

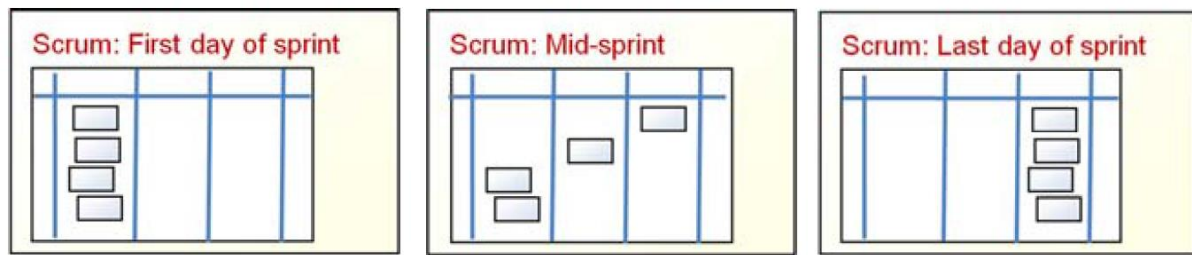


Figure 9. Scrum Board process (Kniberg & Skarin, 2010)

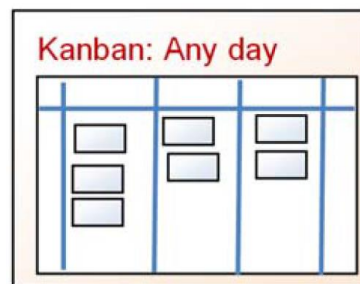


Figure 10. Kanban Board process (Kniberg & Skarin, 2010)

Scrum and Kanban share the concept of being based on incremental development, meaning that you slice tasks into smaller pieces, making the work more manageable. In Scrum the items are split until they fit the time-boxed Sprint. Kanban, in contrast, will let tasks run over a longer period of time but yet have a constant level of flow to minimise lead times, thus naturally reducing items to smaller pieces (Kniberg & Skarin, 2010).

4.6 Hilti V-model / Requirements Engineering

Hilti utilises the V-model as a framework for Requirements Engineering (RE) activities and it is used to:

- Structure and visualise development activities in connection to RE,
- Highlight certain required work products in connection to RE, and,
- Contribute to the establishment of a consistent and integrated development process at Hilti.

The V-model, depicted in Figure 11 allows to visualise RE activities and work on system and subsystem level and thus emphasise that different requirements belong to different hierarchical levels. Furthermore, it allows for vertical (e.g. system → component) and horizontal (requirements for the associated testing) tracing of requirements. However, the V-model is not intended to represent an entire product development process or life cycle.

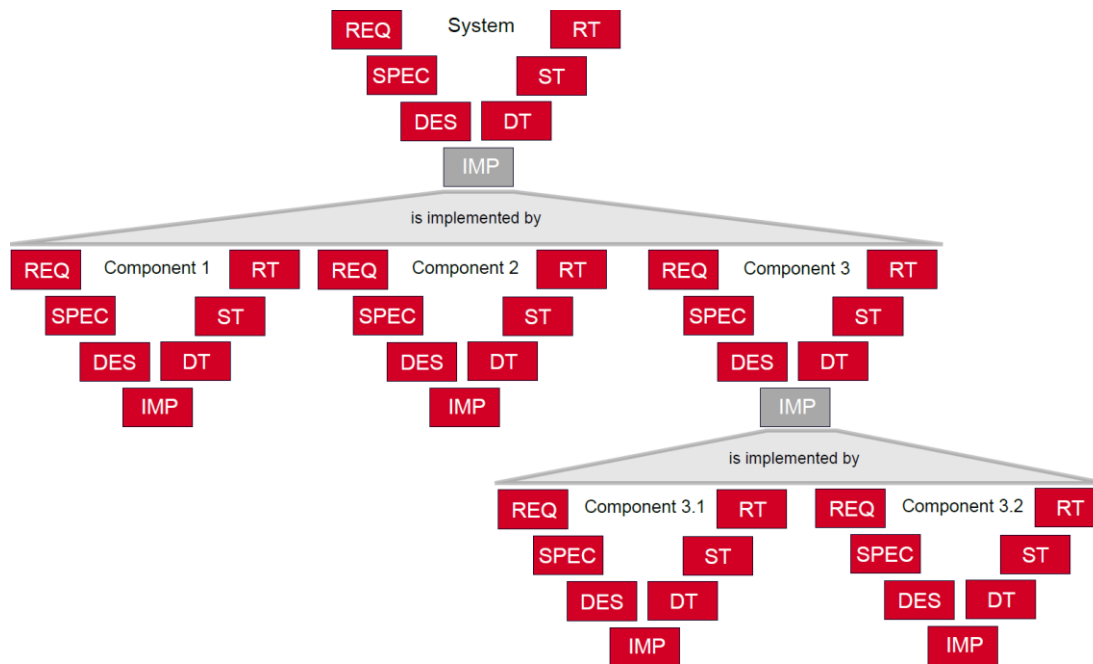


Figure 11. V-model for Requirements Engineering © 2018 Hilti ALL RIGHTS RESERVED

Located on top of the V formation is the **Stakeholder Requirements (REQ)** which is the result from the stakeholder requirement elicitation process. The REQ is necessary in order to have a clear view and common understanding of what the customer wants and the market needs. In addition, the REQ includes business and management goals and legal regulations. The intention for the REQ is to answer *what* is going to be developed but the solution is not included in this level. However, they create a foundation for the next level in the model, which is the **System Specification (SPEC)**.

The REQ is important because they

- clarify the development task in an early stage and include all the relevant stakeholders,
- provide a common understanding about the Targets and support the communication and
- improve the quality when the project is in an early stage and leave space for creativity.

Located on the opposite side of the REQ is **Stakeholder Requirements Test (RT)** which is the process of testing the requirements found in the REQ. The RT consists mostly of customer confirmation of the product that will be developed. Moreover, it includes approval tests which differ depending on the stage of the development and thus different aspects are considered. These tests can be for example Design Acceptance Tests (DATs), Customers Acceptance Tests (CATs) and Customer Confidence Tests (CCTs).

The actual solution space for the development starts from the second level depicted in the V-Model with the **System Specification (SPEC)** and its corresponding testing, the **System Test (ST)**. The SPEC provides a technically feasible system description which has the purpose of a commonly agreed upon system description with measurable product characteristics and provable system tests. The SPEC is therefore intended to detail and quantify what will actually be delivered to the customer. These results are then used for the next level in the V-Model, the system **Design (DES)**.

The SPEC is important because it provides

- a technically feasible and structured system description and
- traceable functional and non-functional requirements

The corresponding test segment on this level is the **System Test (ST)** which works as a system verification and ensures an implementation free from errors. The testing is carried out through sample and prototype tests depending on the maturity. These verifications are then included in the release management plan which in turn determines the order of implementation of the requirements in the system.

Located further down the V-model is **Design (DES)** and the corresponding **Design Test (DT)**. The design is based on the result from the SPEC and identifies which requirements that belong to which elements of the system. Looking at DES on a system level, it provides an architectural design with a System Breakdown Structure. Both the SPEC and DES answer the question *how* to develop the system.

DES is important because it

- breaks down the system into subsystems and provides the development solution,
- allocates solution components to the system requirements and helps to structure projects and thus
- enables efficient cooperation between the component development activities

The **Design Test** verifies the system integration and makes sure that the components are functioning as one unit.

The **Implementation (IMP)** represents the physical and logical combination of the system that together build the complete system.

4.7 The Hilti Technology Process (TP)

Hilti TP projects are conducted to develop and prove that the technology for a product is feasible and has reached a proper level of maturity to start developing the product in its entirety and thus making a transition into a TTM project, detailed in the next chapter. This is similar to Lean Product Development where you ensure that the project can successfully be executed by filling knowledge gaps before too much time and money is invested.

TP projects are split into four milestones (TP1-TP4), as can be seen in Figure 12 where the first milestone is to get the project order and a definition of how the project will be run. The following phase, technology definition, the requirements of the technology to be developed are analysed and pinpointed and finally concluded in TP2 including a project plan. TP3 intends to deliver a requirements specification, assessments of risks and IP etc. and various alternatives are evaluated and assessed. The conclusion of this phase is marked by the selection of one or more alternatives for advanced development. In the technology development phase, the technology is developed and expanded to such a level that it can be shown to be ready for use in product development. The technology project is concluded with TP4, followed by implementation in a TTM project. The goal of a technology project is to minimise risk before the TTM project begins.

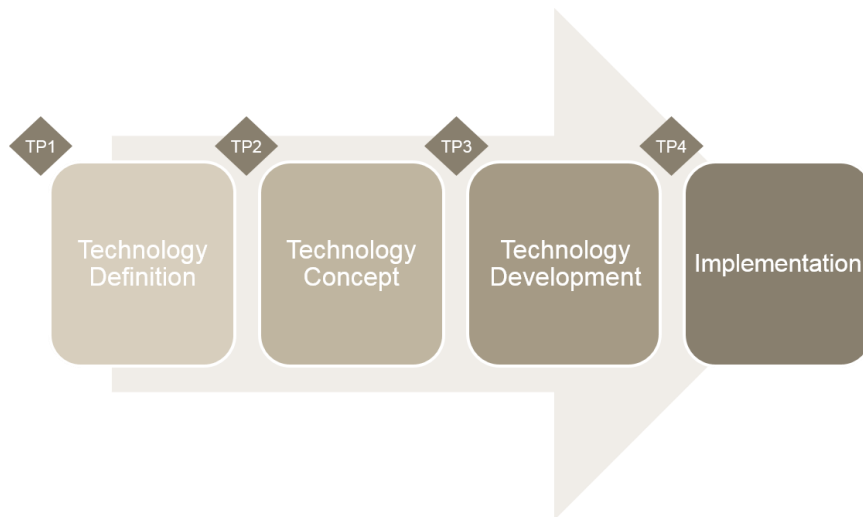


Figure 12. The Hilti TP Process

4.8 The Hilti Time To Money (TTM) process

The Hilti TTM process is a six-gate (G0 is in practice only a kick-off) and six-phase model, illustrated in Figure 13, with each gate having defined key deliverables which are criteria that need to be achieved and presented to be able to move on to the next phase. The timing of when the individual content and activities are to be delivered within the phases are on the other hand not specified, that is up to the teams themselves to decide and plan. In addition to defining all the deliverables, the TTM process intends to outline a common communication frame where everyone “speaks the same language”, knows exactly what to do and give support to management of the project. The Scope of the TTM process is to bring new products and services to the market, starting with a business opportunity, and is concluded with a launch of the new product/service.

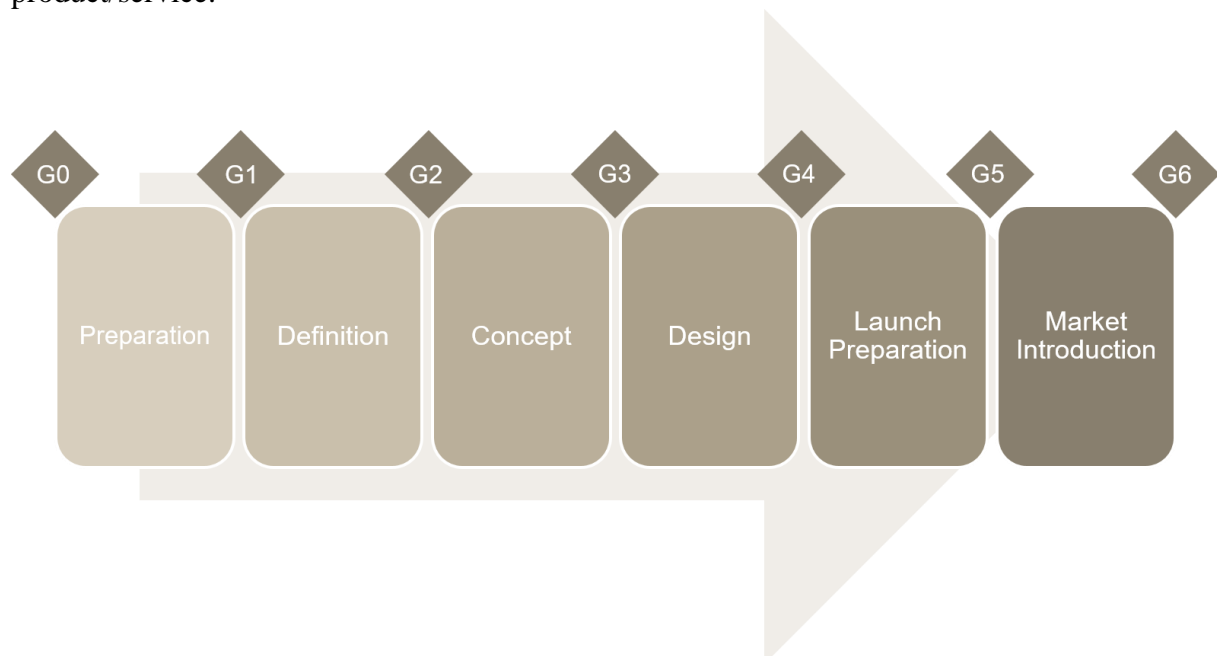


Figure 13. The Hilti TTM Process

G1 deliverables are characterised by making a rough plan (including risk identification and various assessments) until G5 and implement learnings captured from previous TTM projects. G2 includes composing a team with the right competencies and detailed project plan until G3. The projects financial Targets and other project strategies are further clarified and established.

G3 contains project Targets that have been worked out in detail and verification of these. In addition, product specifications are presented and concept and prototype testing approved among many more deliverables. G4 includes testing which needs to be passed. Functions like Repair and Production get the necessary data to start their development. G5 emphasises that the product shall be ready for the market and field and repair tests are passed. The product, including packaging and repair, should be completely ready for operation. G6 is the gate where the product has been launched in the different markets and thus when money is, if things go as planned, generated. Learnings are documented and the project is transitioned into the maintenance phase.

If a meeting room is continuously used by the same team during a TP or TTM process, it is given the name “obeya” which derives from Japanese, meaning “large room” or “war room”. The room is dedicated to the team’s meetings and includes all the material and tools needed during meetings, thus can any team member easily go to the room to look up something project-related looked up.

5 THE HILTI AGILE HYBRID

What makes the Hilti case of special interest and what the author, during the time of the research managed to realise and employ, is the integration of two methodologies, namely the Hilti Phase-Gate and the Agile Framework which together form the Hilti Agile Hybrid, as can be seen in Figure 14.

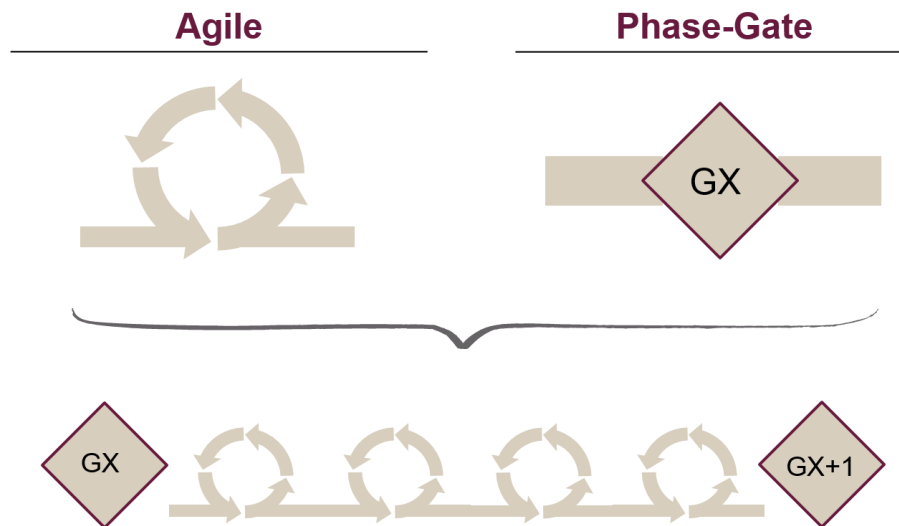


Figure 14. Agile Framework and Phase-Gate combined

The Agile Framework is integrated in between the gates of the Phase-Gate structure, i.e. during the actual execution of the project, while the gates from the Phase-Gate methodology are kept and act as risk-mitigating control points for budget, time, quality, product maturity, the project's progress etc. Each gate represents a number of criteria that need to be fulfilled in order to let the project continue to run.

Much of the mind-set, tools, materials, events, meetings etc. were adapted from literature of e.g. Scrum in addition to having a close contact with consultancy from an external company with over 20 years of experience and fine-tuning Frameworks for both SW and HW development. The author conducted interviews with representatives from the consultancy company to gain further knowledge and spark ideation in the topic. Furthermore, Hilti had several pilot-projects that intended to try out slightly different variations of the Agile Framework in order to find out what is working best and in turn consolidate best practices. Throughout the period of the author's presence in the company, documentation, interviews and observations of people within and outside the company were conducted in order to assess and propose potential improvements. In addition, conversations were held through e-mail and information was gathered with questionnaires to gain further knowledge in certain topics. The outcome was detailed by the author in a thorough document that covers all the aspects of the Hilti Agile Hybrid Framework, explained in "The Hilti Agile Guideline / Documentation", for a future global roll-out and creation of training material for new teams to easily set up their own customised structure that contains all the best practices from pilots that were run.

The TP / TTM Framework that was used among all HW projects when the author arrived to the company was comparatively slow and rigid and was in need of a change to be able to stay competitive and stay in line with the company strategy. With optimism, the Agile

methodology will be rolled out globally with a quick ramp-up, creating synergy effect between the teams.

The Hilti Agile Guideline / Documentation does not have an immediate effect but it will be of help when new teams anticipate to utilise the new Framework. Teams will always be able to reside to the Guideline / Documentation to find whichever information they need to perfectly execute the methodology.

5.1 Pilot Projects

The Pilot Projects all had different characteristics, both in terms of project type and project management processes. The common aspect was that they all had been more or less consulted by an external consultancy company that had a foundation of how the methodology can be conducted.

5.1.1 Pilot project A

Pilot Project A started applying the Agile Framework while being in a TP process but then got transitioned into a TTM process. In that stage, it was decided that since there would be significantly more value / work streams involved it got split up in two where one handled the drive part, i.e. motor, driveline etc., and the other one the system around the product. Information between the two was shared by having a common product owner that could transfer information downstream in the teams.

Pilot Project A was a TTM process between G0 and G1.

5.1.2 Pilot Project B

This project was mainly executed in the same building as the author worked in and thus it was easy to attend several of the team's meetings. By doing so, it was possible to identify small changes from meeting to meeting, which often depended on the team's preferences and needs. For example, some parts of the schedule were removed every second meeting because it would not provide much more new information than what the team already knew since the previous meeting.

Pilot Project B had a characteristic of being slightly less complex in the sense that it did not require as many different experts from several areas. Thus, the meetings did not usually contain a vast number of people.

Pilot Project B was a TTM process between G2 and G3.

5.1.3 Pilot Project C

Pilot Project C is mainly developed in Germany and is of higher complexity. The meetings usually take longer time because there are more value / work streams that need to align their work. The project use JIRA for Target and issue management and a large chart is used to track the overall progress of the project.

Pilot Project C was a TTM process between G2 and G3.

5.2 The Hilti Agile Guideline / Documentation

The guideline /documentation is a walk-through of the entire process of the Hilti Agile Framework. First it gives an explanation of one of the most central aspects of the Framework – the Sprint, which large parts of the methodology build upon. The following chapter deals with all the materials and tools being used by Agile teams. Then, the guideline / documentation covers the different roles and responsibilities that the members and managers have. The following chapter then clarifies the meeting culture, i.e. what kind of meetings exist, how often they are held, the participants and their respective outcomes and then concludes with preparations, such as software and methodology support, that are necessary for a team to start an Agile project.

5.2.1 The Sprint

The Sprint is the heart of the Hilti Agile Framework and is a continuous fixed 4-week cycle that implies risk mitigation and structure for the development work in repeating cycles throughout the phases of the project (TP1-TP4 / G1-G6). It contains all the tools, materials and meetings that will be further defined in coming chapters. The outcome of the Sprint is a potentially shippable piece of the product which is achieved only if the team reaches the Sprint goals. This is planned jointly in the beginning of the Sprint and represents the acceptance criteria agreed upon by the team members and is then assessed at the end of the Sprint. Figure 15 gives an overview of how a Sprint in its simplest form is conducted.

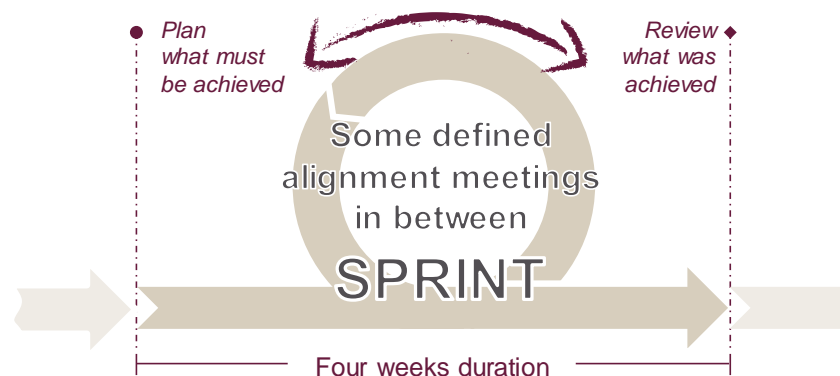


Figure 15. The Sprint © 2018 Hilti ALL RIGHTS RESERVED

Why 4 weeks?

From literature and case studies, the Sprint cycle does not have to be 4 weeks but rather be limited to a maximum of four weeks. However, the author discussed the reasoning behind using 4 weeks for the Sprint cycles with the managers of the pilot projects and concluded that it was more feasible with 4 weeks. The reason is because the framework involves delivering a potentially shippable product by the end of each Sprint but this is proven to be difficult in HW development (2 week Sprints was previously tested) because of higher level of dependencies from different functions, physical suppliers, testing etc. In software on the other hand, results are generated with a lot higher frequency because of lacking physical interactions.

5.2.2 Agile Materials and Tools

Depending on the team set-up, preferences, the project characteristics and its complexity, a different mixture of Agile materials and tools were used and adapted. The purpose of this chapter is to present all the different tools and materials and how they are used, then it is up to

the team to make a decision of which ones that are suitable and ultimately used in their specific project. Table 1 provides an overview of all the different tools and materials that can be used in an Agile project including a brief description.

Table 1. Overview of the Agile tools and materials

Materials and tools	Purpose teaser
Masterplan	<ul style="list-style-type: none"> • Central element to visualize overall project progress (critical paths and timeline until G6) • Detail view of actual Sprint incl. deliverables/ Targets, risks, trend charts, Sprint contract, etc. • Prioritisation, dependency refinement from value stream perspective (Product Backlog) • Belongs to the whole team
Product Backlog	<ul style="list-style-type: none"> • List of “everything” that is known to be needed in the product (value / work streams POT) • Implemented in the Masterplan
Sprint Backlog / Board	<ul style="list-style-type: none"> • Subset of Product Backlog (left side Masterplan) • Project Team to fill out
Kanban Board	<ul style="list-style-type: none"> • Personal “board” to manage tasks derived from the Sprint Backlog / Board of the Masterplan • Track status of personal Targets and tasks to be able to inform team in case of issues
Learnings	<ul style="list-style-type: none"> • Documentation of successes and improvement potentials. The latter are input for the Retrospective • Include learnings from predecessor into Kick-Off
Risk Radar	<ul style="list-style-type: none"> • Matrix to document and prioritise / rank risks (economic, technical, legal) • Matrix scales [low, medium, high] for risk impact and risk probability axis
Sprint Contract	<ul style="list-style-type: none"> • Agreement of the team what they will deliver in the planned Sprint • Description of overall Sprint Target • Includes four parts <ul style="list-style-type: none"> – Sprint Backlog Targets / Deliverables from Masterplan (shall/must Targets and in addition should Targets) – Learnings from previous Sprint to be implemented – Risks from Risk Radar to be mitigated in the Sprint
Decision Board	<ul style="list-style-type: none"> • Documentation of decision makings during Sprint (when, what, by whom, which Sprint and project impact)
Trend Charts	<ul style="list-style-type: none"> • Team Satisfaction voting • Success Probability Project OTD voting • Success Probability Sprint contract voting • Risk Trend Chart
Burndown Chart	<ul style="list-style-type: none"> • Graph that indicates workload planning quality / accuracy in self-responsible Agile teams • Ratio of Sprint Contract Targets achieved vs. left over
Team Onion Model	<ul style="list-style-type: none"> • Flexible Sprint team composition to optimally serve the nature of the complex TP / TTM projects • Agile Core, Extended and Full Team <p>➡ Detailed description in chapter “Agile Meetings”</p>

Masterplan

The Masterplan is a timeline board that visualises the critical path and interdependencies for the whole project. The planning has two levels of detail:

- Draft planning with major Targets until TP4 / G5
- Full details one month ahead (Sprint)

The Targets are placed in the row of the value / work stream (a set of people or expertise in a specific topic) with the corresponding responsibility for the specific Target. The top left part of Figure 16 illustrates how the complexity of the project changes from being a consumable / insert to a mechatronic system. With higher complexity, more effort is required from the different value / work streams including sub value / work streams to plan the workload. The large rows show the main value / work streams:

- Project Management
- Go-to-market Functions
- Product / Technology Development Functions
- Supply / Sourcing / Manufacturing Functions
- Learnings / Knowledge Management

These are in charge by the different roles (TTM PL (Time To Money Project Leader), BU PM (Business Unit Project Manager), TPL (Technology Project Leader), Plant). The Targets are planned jointly with these roles and each specific responsible sub value / work streams / functions within the main value / work stream. These rows form the foundation for the Masterplan itself (to the right in Figure 16). The far right part of the Masterplan shows the draft planning with gates included. The further distance in the future the plan reaches, the less detailed is the plan while the opposite goes for the left part. Furthest left is the Sprint Backlog / Board that shows what will be done in the coming Sprint and is thus a highly detailed four-week plan. The middle section shows more specifically *who* is responsible for *what*. Due to the flexible nature of Agile Methodology, the board is continuously changing in case of rising challenges or new requirements and risks.

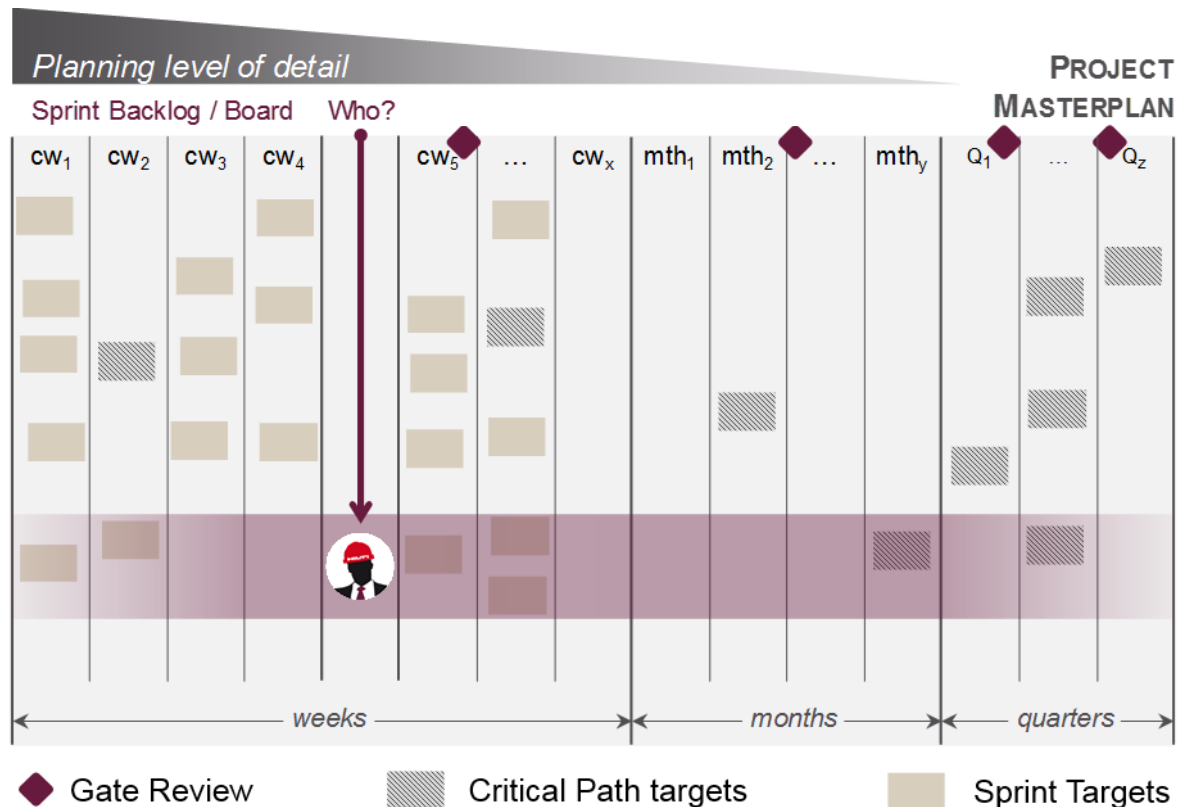


Figure 16. Conceptual Masterplan © 2018 Hilti ALL RIGHTS RESERVED

Why not plan further?

The level of detail in the planning is central while having an Agile mind-set. If there are for example changes in the requirements, unexpected delays or resource capacity variations, it is necessary to be able to keep a stable plan ahead. The goal is to have as little negative consequences on the overall project progress as possible. If a project takes several years until launch, it is very common that these types of changes occur and if the entire plan is already determined from project start until end, small deviations may require a vast amount of changes in many areas of the plan which may in turn cause disturbances to the project as a whole.

The Agile Framework builds on the possibility to take in changes easily. Therefore, the projects are structured in a way to avoid that all deliverables depend heavily on previous work and in that way not having to rework all progress so far but instead just a fraction of it. The method relies on breaking a big goal into Targets which can be worked on concurrently.

Target status

Targets are described on sticky notes which are then placed in the Masterplan and/or Sprint Backlog / Board. Each Target has a specific colour or rotation to indicate its status. Figure 17 shows an example from Pilot Project A where only colours were used to illustrate Target status differentiation. The blue, 90 degree turned square, indicates a key date. Light yellow is a target with regular importance. Dark yellow is of high importance, for example a critical path target. Light green indicate new targets that popped up during the sprint. Pink illustrates a target that have been postponed or that is impossible to achieve during the sprint. A white post-it is a target that have been finished already. Orange shows a target that is absent. Turquoise are targets which were originally planned and later cancelled.

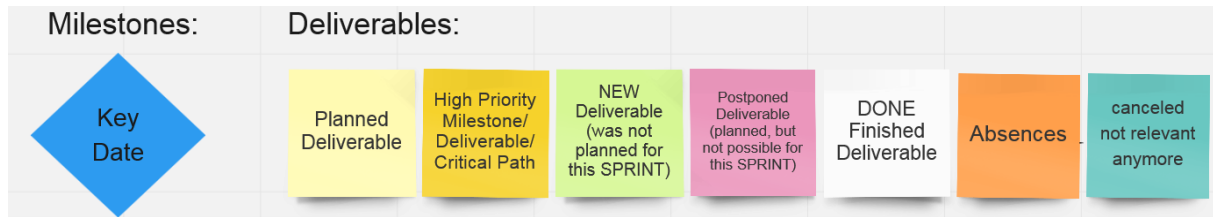


Figure 17. Example of Target statuses © 2018 Hilti ALL RIGHTS RESERVED

Different teams have different needs

It was discovered that not all teams prefer using the exact same scheme of status indications. As can be seen in Pilot Project A, a fair amount of different status indications was used according to a strict plan while Pilot Project B had a slightly looser schedule. The author asked members of both teams about their perception of how clear the indications and meanings of the statuses were. They responded that there were no concerns or misinterpretations in general. The strictness might have both positive and negative impact: with a too rigid structure, the methodology may seem a little bit stiff which is the opposite of what the Agile Framework intends to elicit. However, without structure there might be miscommunications rising.

Product Backlog

The Product Backlog includes all requirements / wishes from an end customer perspective that the BU PM captures as “user stories”. These are then transformed into Targets that are put in the Masterplan, illustrated in Figure 18. The Backlog items are necessary to be fulfilled to finalise the product.

User stories: “the product should have feature X”



The user stories is broken down into targets put on the Masterplan

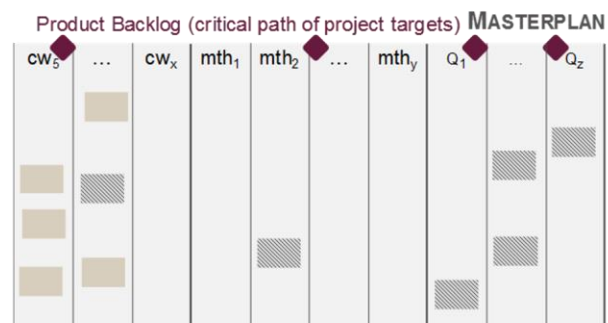


Figure 18. Backlog “user stories” implemented in the Masterplan

New Targets that pop up

If there are for some reason new Targets that surface during the Sprints, they are captured in a specific area next to the Masterplan and then investigated more closely during a joint team meeting where it is decided when or if these are implemented in coming Sprints.

Sprint Backlog / Board

The Sprint Backlog / Board is a board next to the Masterplan that builds the basis for the joint team creation of a Sprint Contract. The board visualises the product Targets for each value / work stream which were pre-defined by the Product Owner Team and adjusted by the Project Team to finally conclude the Sprint Contract. Furthermore, it highlights Target dependencies

of the upcoming Sprint, which is conceptually illustrated in Figure 19 by using a red line. The columns depict the week of the Sprint (1-4).

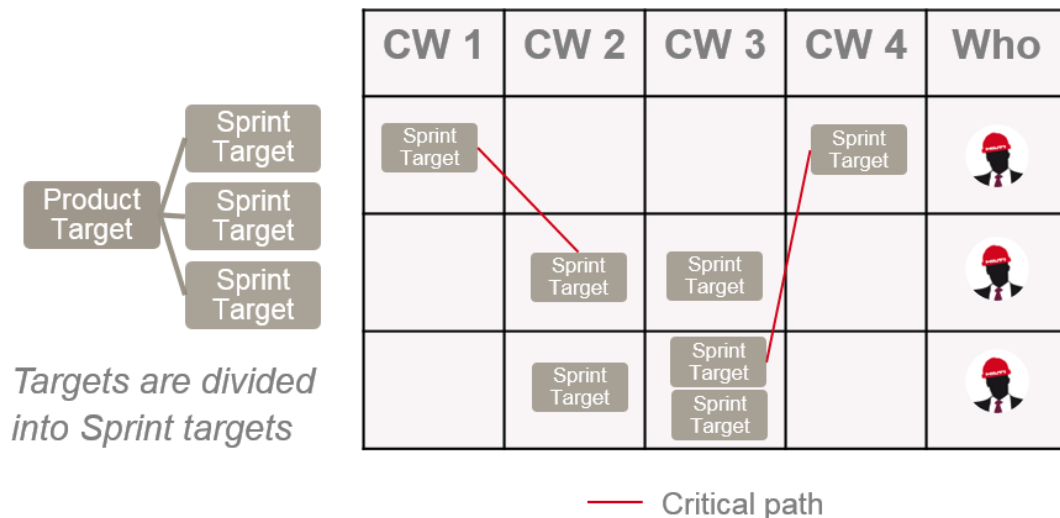


Figure 19. Sprint Board / Backlog

From a Target being on a higher level, in the picture above called “Product Target”, it is broken down into smaller Sprint Targets which are then properly aligned according to how the value / work stream will address the Targets.

Teams can choose to illustrate the critical path in their own way

Some teams use different colours on the sticky notes to elicit the critical path, others use lines between Targets.

Kanban Board

While Targets in the Masterplan are drafted, the Targets in the Sprint Board are further well-defined. The realisation of *how* to reach the Targets is however not specified. That is up to the individual team member to clarify with the Kanban Board (see chapter “Kanban”). The Kanban board works as a personal task managing tool where the Sprint Target is broken down into well-defined tasks, illustrated in Figure 20. If all the tasks are finalised, the Target is reached.

Sprint targets	To do	In Progress	Done / To verify	Closed/ Resolved
Sprint Target	Task Task Task			Task
Sprint Target		Task	Task Task	
Sprint Target	Task	Task		Task Task

Figure 20. Conceptual Kanban Board

The individual Kanban boards are visible to the whole team, which creates a very transparent way of working. The tasks move from “To do” to “Closed/Resolved” depending on the tasks’ status.

Empowered team members

The individual team member is empowered to self-responsibly manage his/her own planning and achievement of his/her targets. Where and how he/she visualises it is up to himself/herself but it is recommended to use a common software, like JIRA.

The author asked team members how they feel about the increased self-responsibilities. Most are positive to it, meaning that a higher level of trust is induced. Some prefer being told exactly what to do, i.e. not planning anything, only executing.

Learnings and Successes

Learnings and successes are central to capture in Agile teams and that can be done in different ways; either during an explicit Sprint meeting that focuses on assessing learnings / successes *or* in the midst of a Sprint. Learnings and successes may concern any aspect of the project or process. Exactly what tool used by the team to capture the potential learnings and successes can be different in different cases.

2-by-2

The 2-by-2 chart is another type of learnings board that is used according to the schedule depicted in Figure 21. The POT brings up one or two subjects that they think are necessary to discuss. It could be Project (assigned round 1) and Process (assigned round 2) which are discussed in two separate sessions in the team. The left side of Figure 21 depicts the amount of minutes for each stage of the assessment paired with a description. The right side shows an example of how the 2-by-2 chart looks like in Pilot Project B.

5 min	Every Team member writes 2 sticky notes with 1 action idea (that he/she feels should be in the Sprint Contract) each that the team should implement in the next sprint
15 s (2 min)	Every Team member pitches each idea for 15 seconds
2 min	Each Team member adds his/her name to the sticky note that he/she finds most inspiring
1 min	Take off the ideas that did not receive any names
1 min	Rate the notes according to number of names on them
5 min	Decide which ones you want to implement and add them to the Sprint Contract. The ideas that you want to keep are put in the Learnings Backlog

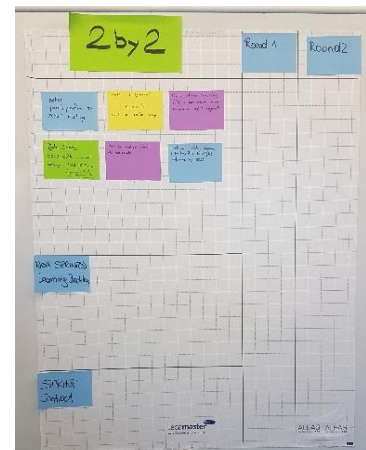


Figure 21. Description of 2-by-2 chart

Learnings & Successes Board

In order to capture what went well or what needs to be improved in the Sprint, in both the process and project, a dedicated board can be used. A conceptual board is illustrated in Figure 22. The team members use sticky notes on which they write their experiences during a joint

team meeting that they then put on the board. Depending on the character of the success / improvement, they may be implemented in the Sprint Contract for the coming Sprint.

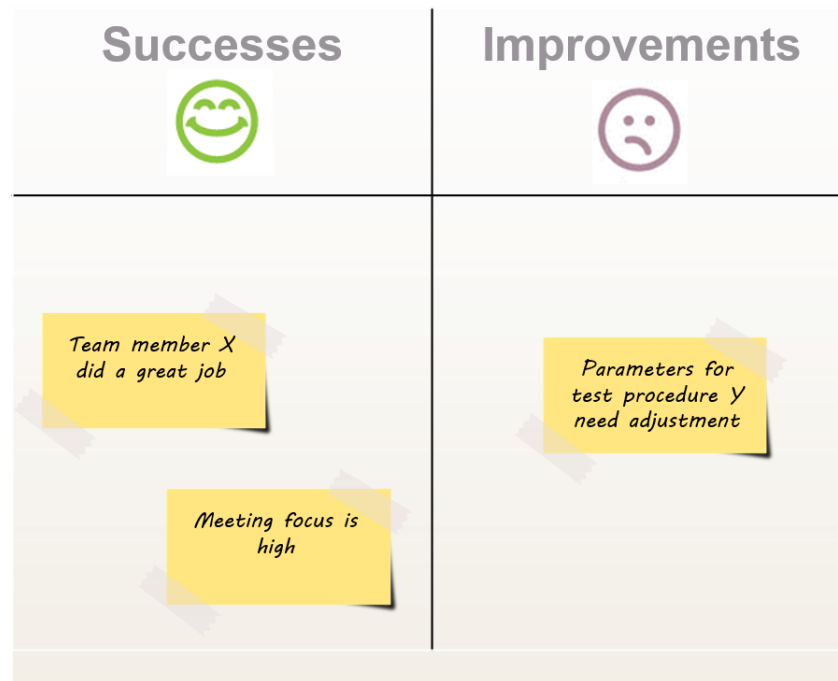


Figure 22. Conceptual Successes / Improvements Board

Successes / Improvements during Sprint

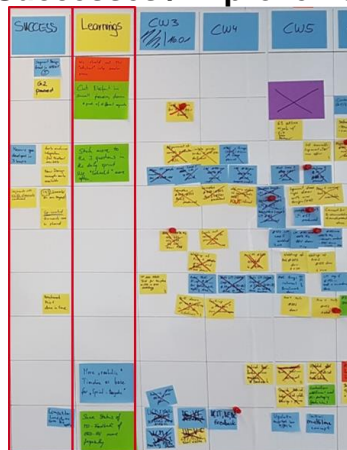


Figure 23. Successes / Learnings in Pilot Project B

To easily capture learnings / successes *during* a Sprint, Pilot Project B came up with the idea to have a separate column right next to the Sprint Board, as can be illustrated in Figure 23. Any team member can then, in any type of meeting, take a sticky note and write his/her experience and put it in that column. These items are then thoroughly assessed in a joint Sprint meeting that includes all members so that everyone can take part of the information and consider the successes / learnings to be implemented in future work.

Visualisation of the successes / improvements

As the teams evolved and grew into the methodology, they made more and more changes to their own preferences. To boost the team's motivation, some teams put some extraordinary successes permanently and highly visible right next to the Sprint Board so that it could always be viewable while interacting with the Sprint Board. That way the successes were not just something that was mentioned once and then forgotten but rather a permanent cheer for everyone involved. The same technique was adapted for learnings / improvements, e.g. that

the team should keep one of their time-boxed meeting to 15 minutes without trespassing that limit.

Risk Radar

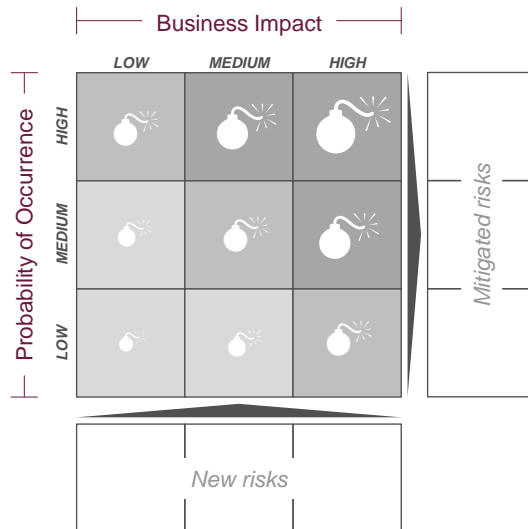


Figure 24. Conceptual Risk Radar © 2018 Hilti ALL RIGHTS RESERVED

The Risk Radar, shown in Figure 24 as a conceptual image, works as a tool to document, highlight and rank the different risks that the project faces. The board is a 3x3 grid with three different areas, represented by three shades of grey. The team takes stickers with the risk written on them and then puts them on the respective area depending on how severe it is perceived to be. The x-axis ranks the Impact and the y-axis the Probability. The Risk Radar covers any issues related to the project, such as financial, production, suppliers resources etc.

In Pilot Project A, the risk radar had colours ranging from yellow (Low) to orange (Medium) to red (High), as can be seen in Figure 25. The green area indicates that the risk has been completely eliminated and does not anymore pose a risk to the project / process. In addition, the stickers can also have different colours depending on the risk's attributes, for example: an orange sticker could indicate a risk that poses a threat to the critical path. Yellow denotes if the risk is related to Targets with normal priority. Risks written on green stickers indicate that the risk is newly discovered for the project.

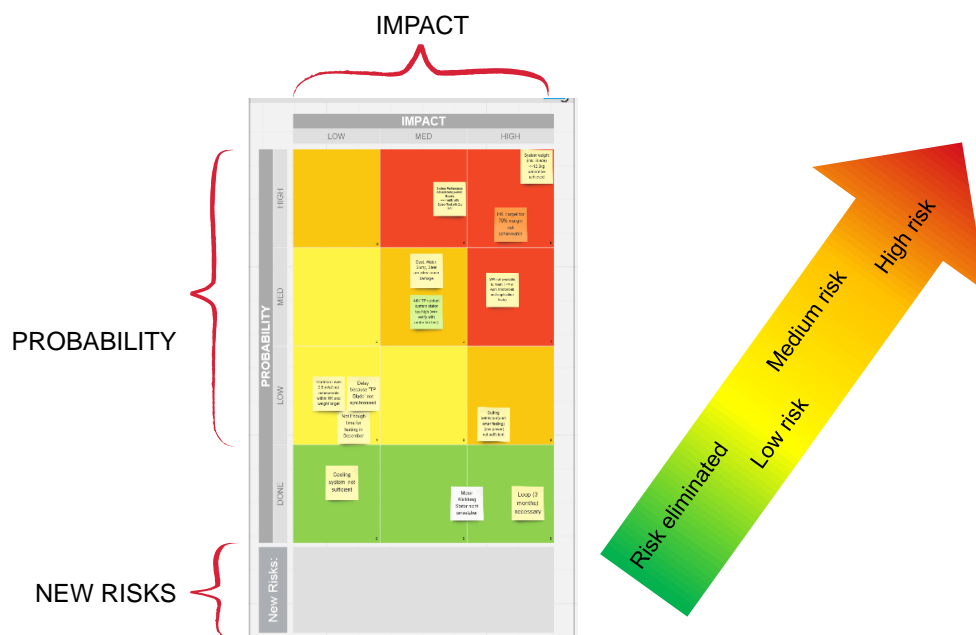


Figure 25. Risk Radar used in Pilot Project A

Depending on the team's perception of the item's Probability and Impact, the item receives a spot on the board and indirectly a colour associated with a severity number (green = 0, yellow = 1-2, orange = 3, red = 4-5) which are in turn calculated using a template. The Risk Impact Value (RIV) is assessed once every Sprint and tracked throughout the project. The RIV is calculated according to the following schedule which explains the process shown in Figure 26

1. Count the number of stickers in each "zone" (red / orange / yellow / green)
2. Multiply respective zone's number with the weighting for that zone
3. The result per area is displayed and a total result is shown
4. The total result is tracked in the Risk Trend chart.

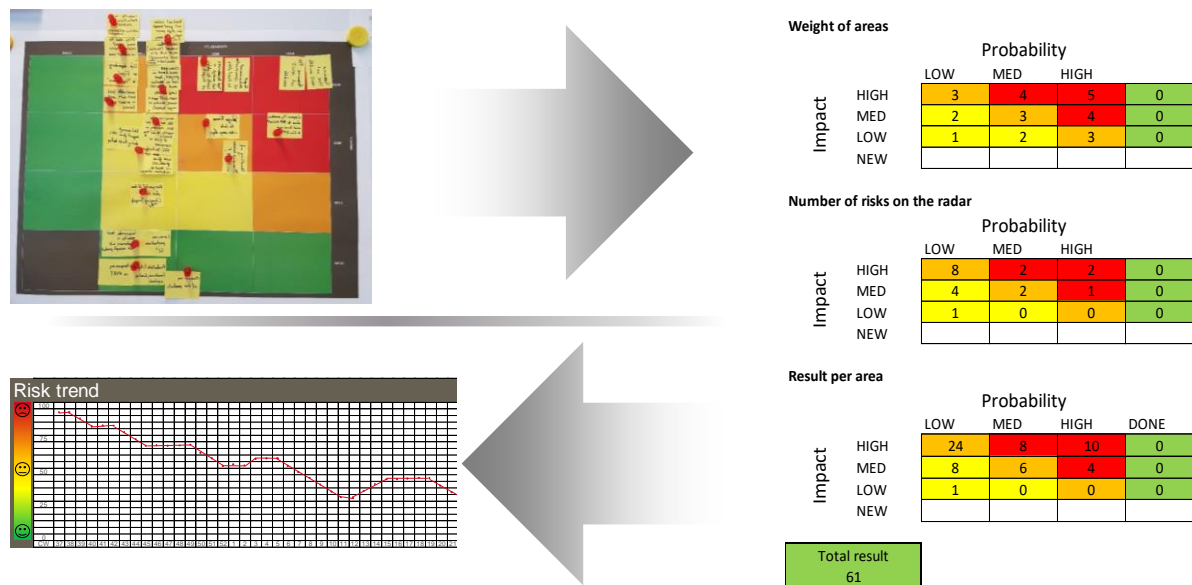


Figure 26. Risk Impact Value calculation and tracking © 2018 Hilti ALL RIGHTS RESERVED

Why a Risk Radar?

The Risk Radar is important because it highlights potential hurdles that are in the way of reaching the Targets. Having highlighted the risks, it is possible to create a mitigation plan for them and thus minimising the risk's severity. The ambition with the risk trend is for it to decrease over time as risks get diminished, however, if new risks pop up or some risks get their probability or impact increased because of previous misjudgement, the trend might rise.

Sprint Contract



Figure 27. Conceptual Sprint Contract

The Sprint Contract is a document resulting from a joint team meeting, and works as a contract that the whole team agrees upon. Based on what the team identifies as the most important Targets for the Sprint (i.e. what characterises the Sprint and the outcome), a title that represents and characterises the Sprint is given to the contract. The segments of the contract are Targets (MUST and SHOULD), Learnings Implemented and Risks Addressed, as can be seen in the conceptual image in Figure 27.

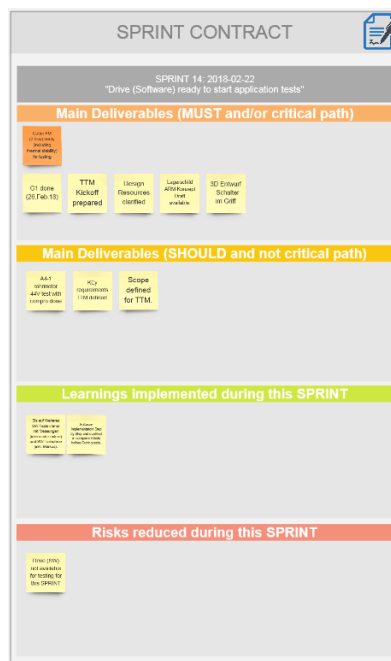


Figure 28. Sprint Contract used in Pilot Project A

Pilot Project A had a slightly different structure and naming for the different headings in the contract but the content remains the same. Instead of paper based, the contract was a part of the online tool RTB, illustrated in Figure 28.

The construction of the Sprint Contract

What is being put in the Sprint Contract is a mixture of subjective interpretation and decisions based on what the coming Sprint was planned to cover. The sequence is initiated by looking at the Masterplan and slicing the Masterplan Targets into Sprint Targets and then choose the most important Targets for that particular Sprint to be put in the contract. The same idea applies to Risks Addressed, i.e., the risks that are most relevant for the activities in the Sprint are put in the contract. The Learnings Implemented could be items that are relevant to either the process in general or project specific improvements.

Decision Board

The Decision Board provides a structured Framework to capture decisions made during meetings. The board should specify in which Sprint the decisions were made, a short description, the reason, team members that participated and which date the decision was made. This will enable easy tracking and possibility to look back to decisions if questions arise. A conceptual Decision Board is shown in Figure 29.

Decision Board			
<u>Sprint:</u>	SP 5	<u>From:</u>	DD-MM-YY
		<u>To:</u>	DD-MM-YY
Description of Decision	Reason	Participants	Date
Change of Supplier for part X	Price not justified	Lars Fredrik Benny	DD-MM-YY
Postponed deadline Y	Delay from function Z deliveries	Lars Fredrik Benny Karl	DD-MM-YY

Figure 29. Conceptual Decision Board

Documentation of decisions

The way that the decisions are documented differ from project to project. Some teams decided to take a photo of the board each time a new decision was taken and then upload to a common shared folder while another team used text editing programs.

One team member noted that a board for documenting decisions was good as long as the importance in the decisions was kept at a reasonable level to avoid too much “paperwork”.

Trend Charts: Burndown Chart, Risk Burndown Chart, Team Satisfaction & Probability of Success

The trend charts are used as performance indicators. The Agile methodology emphasises constant improvement and makes sure that the team can always take learnings. The Hilti Agile Methodology includes quantitative measures and (Burndown charts) and qualitative (Team voting charts). These are regularly updated in joint team meetings.

The quantitative charts are calculated by comparing the number of items that have been achieved / diminished to the original amount. For example, the Sprint Board has a pre-defined number of Targets which are planned to be achieved by the team. That number is then compared to what had actually been achieved. For example, if the Sprint Backlog contained ten Targets and by the end of the Sprint, eight were achieved, it means that an 80 % Burndown rate for that Sprint was reached. If the charts surpass a certain pre-defined threshold, it is necessary to raise the question why that happened. Figure 30 shows a conceptual chart and how values are tracked through several Sprints, including a threshold.

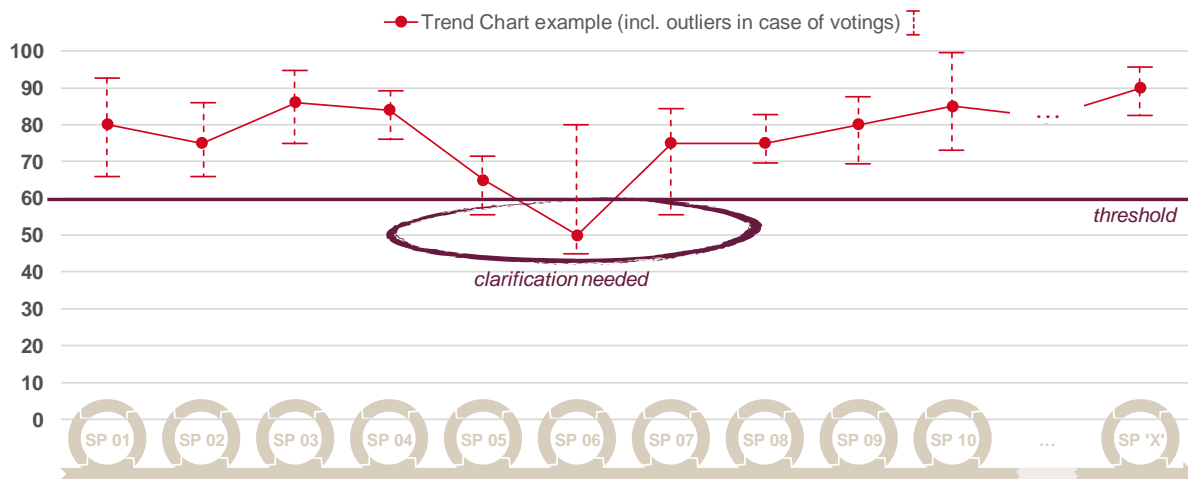


Figure 30. Conceptual Trend Chart

How the chart input is generated and examples from Pilot Projects

Probability of Success for ACTUAL SPRINT	Probability of Success for G5	TEAM Satisfaction Level
100%	100%	Excellent
75%	75%	Very Good
50%	50%	Ok
25%	25%	Poor - I do not like to work here
0%	0%	Poor - I want to quit this team

Figure 31. Team Voting template used in Pilot Project A

The measurements are made by allowing the team members to put a marking on a scale, ranging from 0-100, that corresponds to their perception of how likely it is that they will reach the Sprint Targets, On Time Delivery (OTD) for the project (G5) and Team Satisfaction. After all team members have placed their voting, a mean value is calculated which is the value that is put in the Trend Charts. Exactly how the voting template looks like is individual to the team's preferences, it can for example be a colour-scale ranging from red to green instead of pure numbers like what is shown in Figure 31 to the left.

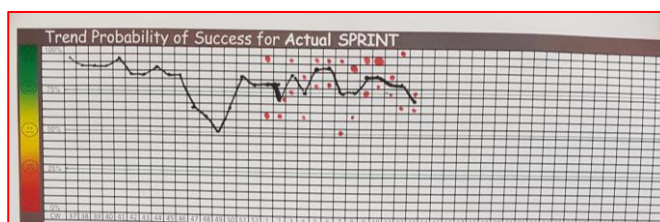


Figure 32. Probability of Success actual Sprint trend

Probability of Success (Sprint) Trend

The team members assess what they think the probability is for the Sprint to reach the Sprint Targets. The result is tracked, as can be seen in Figure 32. If any team member does not think that the Sprint Targets are reasonable, he/she will put a sticky note with his/her thoughts and/or directly discuss with the team about why he/she thinks they are unreasonable.

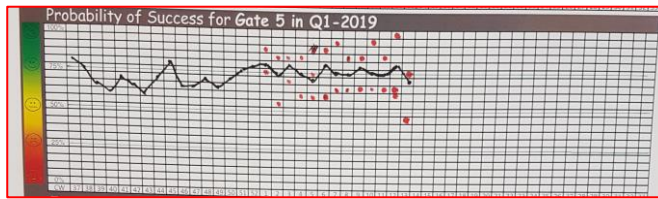


Figure 33. Probability of Success Gate 5 trend

Probability of Success (G5/6) Trend

This trend chart shows how the team members perceive the probability of gate 5/6 being reached within the specified time-schedule. This measurement is tracked through the whole project and an example is illustrated in Figure 33. These numbers are measured every week.

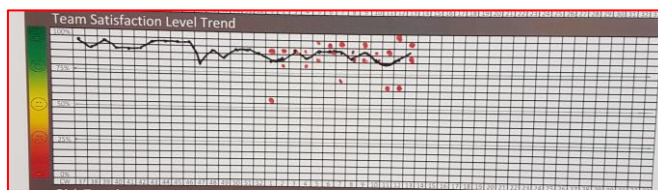


Figure 34. Team Satisfaction trend

Team satisfaction Trend

The team satisfaction chart to the left in Figure 34 is used to identify how the team has perceived the progress of the process and the project during the last Sprint. The level of satisfaction is tracked and if there is something out of the ordinary, it is highlighted and discussed.

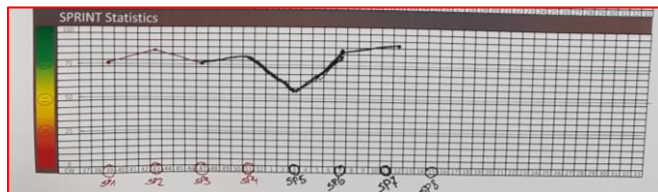


Figure 35. Sprint Burndown trend

Sprint Burndown Trend

Figure 35 shows an example of the Sprint Burndown chart. It measures how many Targets in the Sprint that were achieved in comparison to the number of planned Targets.

What is in it for the team?

The charts have the capability of surfacing any abnormalities in a structured way and gives the team a chance to work on continuous improvement which is a core aspect of the Agile Framework.

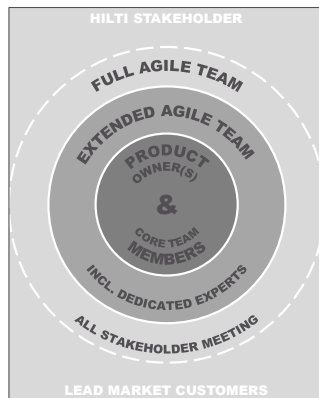
The probability of success reflects the level of confidence of the team members of how likely it is that they will reach the set Targets. If it for some reason is very low, it might be necessary to discuss and take the issue to a higher priority level and/or adjust resources to cope with the challenge.

Team satisfaction level adds another dimension of how well the members perceive how the project is going process wise, in management and the general feel for the progress and team.

The Sprint Burndown chart is important in order to gain an insight into how the team manages to plan reasonably. However, since Targets do not have any weighting or measurable points that reflect on the effort needed, an unfair judgement of the result might be generated. For instance, if one Target is “finished CAD model” while another is “sent e-mail”, these two Targets clearly require different efforts to be reached. Thus, measuring the

number of Targets reached can be disillusioning. This is the reason why Pilot Project C stopped using the Burndown Charts.

Team Onion Model



To fit the nature of HW Development Projects, a flexible meeting culture according to the specific needs of critical resources in various phases of TP and/or TTM projects is needed. Adapting to Agile involves a structure depicted in the Onion Model seen in Figure 36. Depending on what information is necessary to share and decide upon, different people/functions are needed and as the project evolves and moves into different phases, the team roster might be adjusted. The different roles shown in Figure 36 are further explained in the Roles and Responsibilities chapter.

Figure 36. Conceptual Team Onion Model © 2018 Hilti ALL RIGHTS RESERVED

The onion model depicted below in Figure 37 is constructed for an actual Sprint in Pilot Project A. Each sticker has a name of a person/stakeholder/representative etc. in the project. Yellow indicates the Product Owner Team (POT) (further detailed in Roles and Responsibilities chapter), who will attend every meeting and is thus inside the smallest circle, the very core. Green marks the Project Team and it can be seen that the members are scattered with some being in the Core Agile Team while others are in the Extended Agile Team. The reason for this is that some functions / representatives are more relevant than others during this particular Sprint. Thus, the division can change throughout the project depending on what the focus of the Sprint is. The outer onion layer is the Full Agile Team which includes all other stakeholders in addition to the Core Agile Team and Extended Agile Team.

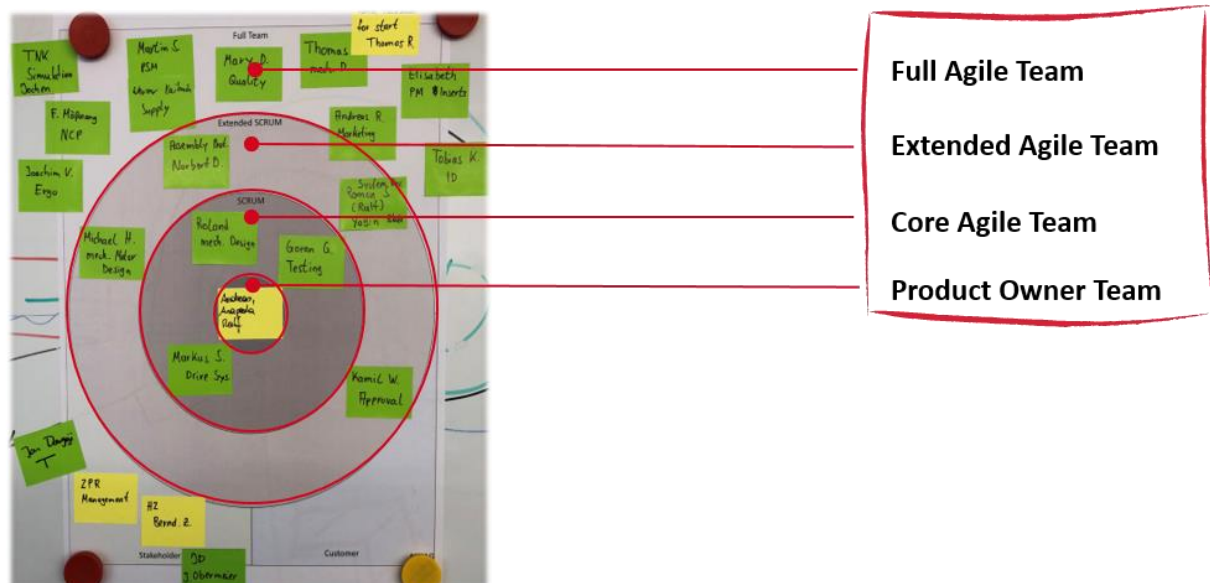


Figure 37. Team Onion Model used in Pilot Project A

Why not do it like regular Scrum?

In Scrum, all team members meet every day to share information, synchronise and allocate resources. This is possible because all members share a similar skillset and thus have a high degree of skillset overlapping, illustrated to the right in Figure 38. This allows for the team

members to easily get an understanding of each other's work and help each other out. In HW development it is more difficult since the skillset of the team members deviate from each other more and their skills are needed at different stages of the product's maturity. For instance, it is impossible for someone responsible for CAD to give valuable input to someone in Quality. The HW skillset overlapping is illustrated to the left side of Figure 38

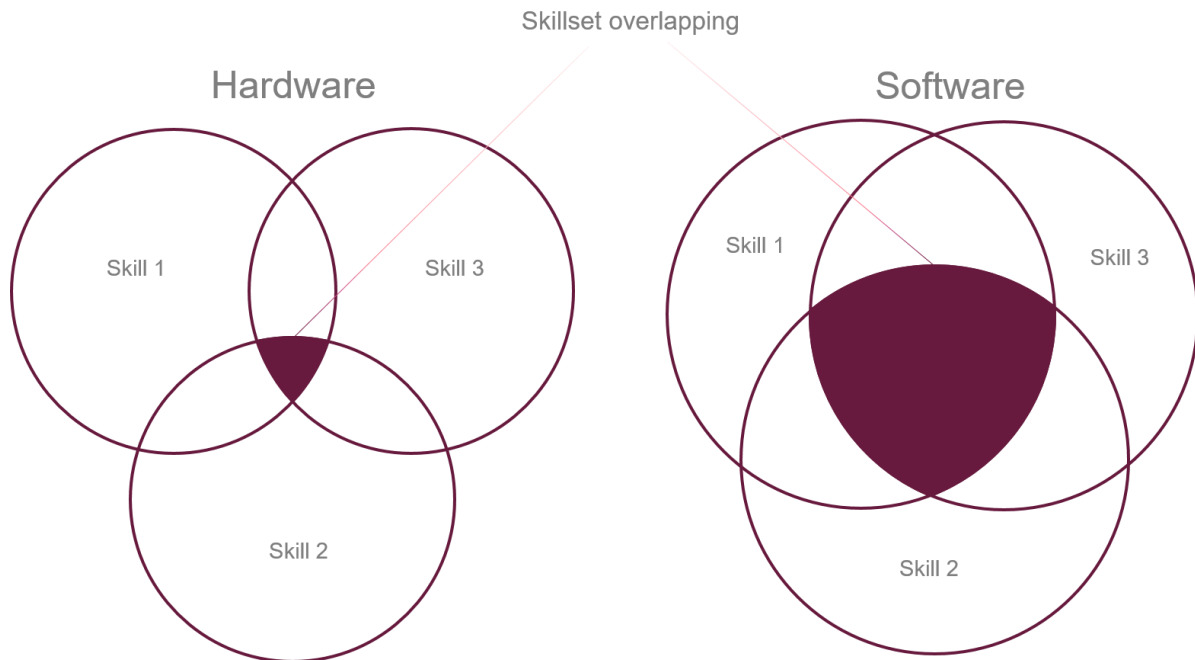


Figure 38. Difference in skillset overlapping HW vs. SW

5.2.3 Roles and Responsibilities

Like in Hilti's already well-established work methodology characterised by Phase-Gate, the project is executed by a project team with defined roles and responsibilities. In a TP project, there exists a *sole* Product Owner whereas a TTM includes a *team* of three Project Owners, a.k.a. the Product Owner Team (POT). In addition, the Agile Project Manager and the Project Team are roles in the Hilti Agile Hybrid.

Product Owner (Team)

The Product Owner(s) is responsible for planning the Targets together with the Project Team, i.e. make a pre-definition of what must to be achieved during the Sprint and throughout the project's life-span. This is then documented in the Sprint Contract. In a TP project it is only needed to have one product owner because the focus is very narrow. A TTM project has a team because they need to align several perspectives. The POT typically consists of the TTM PL to be project accountable, the BU PM to represent the voice of the customer and the TPL to represent the voice of Engineering.

Agile Project Manager(s)

Similar to classic project management there needs to be someone responsible for ensuring project progress, making sure the content is being worked on, resolving team issues, being in charge of all the material included in the Hilti Agile Hybrid, facilitate and plan Agile Meetings and plan for resources and capabilities. The role to ensure Agile project execution is taken over by the respective Project Manager. In a TP project, it will be the TPL while in

TTMs it might be either TTM PL or TPL. In classic Scrum, this role would be called Scrum Master.

Project Team

It is the Project Team that executes tasks and provides the actual project progress. The team is composed of members from the BU, cross-companies and corporate functions. Since each Sprint varies in content, the needed capacity and composition of the Project Team varies as well, resulting in that not every team members needs to take part in every Agile Meeting.

The individual Project Team members handle their tasks by themselves and decide how they will work on the project progress in order to deliver the defined Targets that have been set together with the PO / POT. The tasks are recommended to be handled with a Sprint Board / Kanban Board.

Empowered team member creates trust

Since each team member is responsible for planning their own tasks, an increase in trust has been identified among the members. It is reasonable that the person in charge of his/her own specific area of expertise has the deepest knowledge regarding how much time is needed to complete the Targets. Then it is up to the managers of the project to give the team member the trust that he/she handles the planning correctly.

Responsibilities

The Agile hardware project management roles have the following responsibilities following the RACI scheme:

- Responsible** ▶ The team member who is in charge to complete the Target / task
- Accountable** ▶ The one who approves the completed task and who is accountable for its correctness
- Consulted** ▶ A team member, whose opinion is needed for the completion of the Target / task
- Informed** ▶ A team member who is updated on the result of the Target / task

- **TP Projects:**

Due to origin of nature, TP projects are highly focusing on one core value stream – technology development. Therefore, the TPL is sole Product Owner, being responsible for the overall system, its scope and quality of system maturity. By default, the TPL acts as Agile Project Manager but can delegate this role to any other team member. The responsibilities for TP Projects are visualised in Table 2.

Table 2. Responsibilities TP Projects

Function	TPL	TP Team Member	TP Team
Role	Product Owner Agile Project Manager		Project Team
Responsibility	A, R	R, C	R, C, I

- **TTM Projects:**

TTM projects address multiple value / work stream and thus the main responsibilities vary according to the various phases. TTM PL, TPL and BU PM form the POT, the TTM PL is finally accountable for decision making. The responsibilities for TTM Projects are visualised in Table 3.

Table 3. Responsibilities TTM Projects

Function	BU PM	TTM PL	TPL	TTM Team
Role	POT	POT Agile Project Manager(s)	POT	Project Team
Responsibility	R, C	A (in general), R	R, C	R, C, I

Everyone has got responsibilities

It is important that everyone included in the project feels that they have responsibility to boost the trust between each other. Thus, the project has a feeling of a flatter structure, according to members in the Pilot Projects, where the progress is a team effort instead of the perception of an invisible wall that can in some projects be created between managers and Project Team.

5.2.4 Meeting Culture

An Agile Hybrid project always starts with a Kick-off which works as an initial planning for the project including education regarding the methodology and a chance for the team members to get to know each other before starting.

During the Sprints, there are five types of meetings which all serve their own purpose in a structured way. One type of meeting is held on a daily basis, another on a weekly basis and three on a monthly basis in one blocker (pre-defined time blocked in the schedule).

Kick-Off Meeting

The Kick-Off Meeting is the initial meeting and a one-time occurrence to start the Agile Hybrids. It needs to be prepared for a successful execution. The preparation is similar to waterfall project execution – with the difference that it is necessary to organise Agile Materials and Tools, training and coaching support. The Kick-Off is separated into a kick-off preparation and the conduction of the kick-off itself with the Full Agile Team.

The Kick-Off *preparation* is conducted by Agile Project Manager(s) and Product Owner Team (in general the TTM PL, TPL and BU PM). The Kick-off preparation has the overall objective to result in a pre-filled Masterplan (critical path and Targets), pre-filled risks, team setup and finally a pre-filled Sprint contract.

The Kick-Off preparation will typically last for half to a full day resulting in:

Kick-Off Preparation	<ul style="list-style-type: none">• Clear scope of the project (to be presented by Agile Project Manager in Kick-off)• Introduction to Hilti way of Agility (Materials and Tools, Roles and Responsibilities, Agile Meetings)• All stakeholders defined (Agile Team roster, Lead customers, Project sponsor, etc.)• Resource and capacity allocation committed by BU Management and aligned with cross-company functions• Project infrastructure successfully set up<ul style="list-style-type: none">– Project room available / needed– All meetings (blockers) booked in advance– Decide on physical vs. software material (depended on team roster locations)• All templates available (PowerPoint, Excel, OneNote, Project SharePoint, JIRA, RTB, etc.) for Full Agile Team• Detailed Agenda to kick-off the Agile Hybrid jointly with the Full Agile Team members
U	Pre-filled Masterplan, Risk Radar, Sprint Board / Backlog, Sprint Contract, Team Onion Model, etc.

The Kick-Off itself is a full day event, explaining the Hilti Way of Agility and its relevant materials and tools to the team. This includes:

Kick-Off	<ul style="list-style-type: none">• Rough introduction• Explanation of scope, Targets, key requirements, boundary conditions, etc.• Hilti Way of Agility → Explanation of pre-filled “Agile Materials and Tools”• Conduction of the Planning Game (further detailed later) in their various value / work streams
U	Team committed to refined Masterplan, Risk Radar, Sprint Contract, Team Onion Model, etc.

What is the critical path?

The critical path is essentially the project’s Targets / deliverables that have to be achieved in order to continue the development. For instance: if Target A has contents that are critical to start working towards Target B, then there is a critical path from Target A to B. Within a phase, an example could be a supplier that needs to deliver certain parts before it is possible to continue and if the supplier fails to do so, it may be necessary to adjust the Masterplan.

Meetings throughout the Sprint

There are five types of different meetings throughout the Sprint

- Sprint Planning
- Daily Agile Meeting
- Extended Agile Meeting
- Sprint Review
- Sprint Retrospective

The sequence of the events given as an example throughout the Sprint Cycle are illustrated in Figure 39. The official start of a Sprint is the Sprint Planning, which is the event when all the planning for the Sprint occurs. Then every day throughout the Sprint there is a Daily Agile Meeting for information sharing. Once every week there is an Extended Agile Meeting where the relevant team members share information, discuss content and synchronise work. The Sprint ends with a Sprint Review and Retrospective which intends to inspect what has been done, take learnings and make improvements for future Sprints.

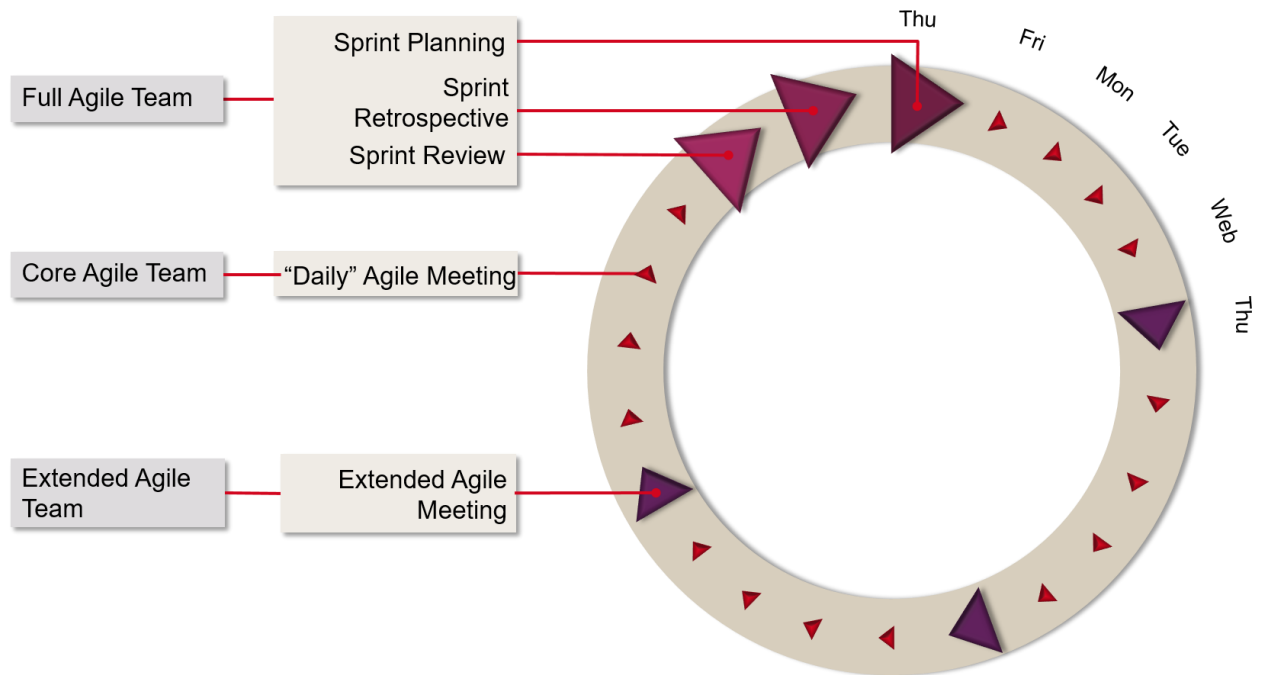


Figure 39. Conceptual Sprint Cycle

The Full Team meetings are held once a month when all team members and other stakeholders, like the customers, sub-contractors and colleagues from outside Hilti, are assembled (except when the meeting enters the subjective surveying phase about the team, project and processes). This meeting includes a lookback at the previous Sprint and a walkthrough of the results generated from the Sprint which are then presented to the team and POT which in turn decide if the Target is met. During these meetings, data is gathered to track successes, learnings, satisfaction, identify risks and take vital decisions for the development ahead. An agile project is intended to be flexible and can thus take new requirements and create changes in the plan in contrast to typical rigid waterfall methodology. In addition, all the data gathered is used for one of the core principles of Agility – continuous improvement. If something did not go well in either the project or the process, it is discussed and documented so that the team can diminish the probability/impact for the same thing to happen in the future. In contrary, if something was executed successfully, notice is taken and the learnings documented and shared in the team.

Sprint Planning

The Objective of the Sprint Planning meeting is to make a detailed plan of the next Sprint (incl. Sprint Contract and adjusted Team Model) and assure that the Sprint Contract is reasonable. Thus, it is important to define Targets that can be achieved within one Sprint which in most cases implies that TP and TTM deliverables need to be sliced into digestible pieces. The Targets must be precisely formulated so that every Agile Team Member has the same understanding of Target content and related effort.

The Sprint Planning is conducted in a two-step approach working with the Masterplan. The first step is a preparation by value / work stream representatives, at least by the Product Owner Team, and has the objective to create a “rough” Masterplan resulting in pre-defined Targets, risks and a pre-filled Sprint Contract. The second part starts with an explanation and discussion of the pre-filled Masterplan jointly in the Full Agile Team. The team members will then refine the plan according to their expertise leading to a final and jointly committed Sprint setup as illustrated in Figure 40.

PLANNING GAME – A TWO STEP APPROACH

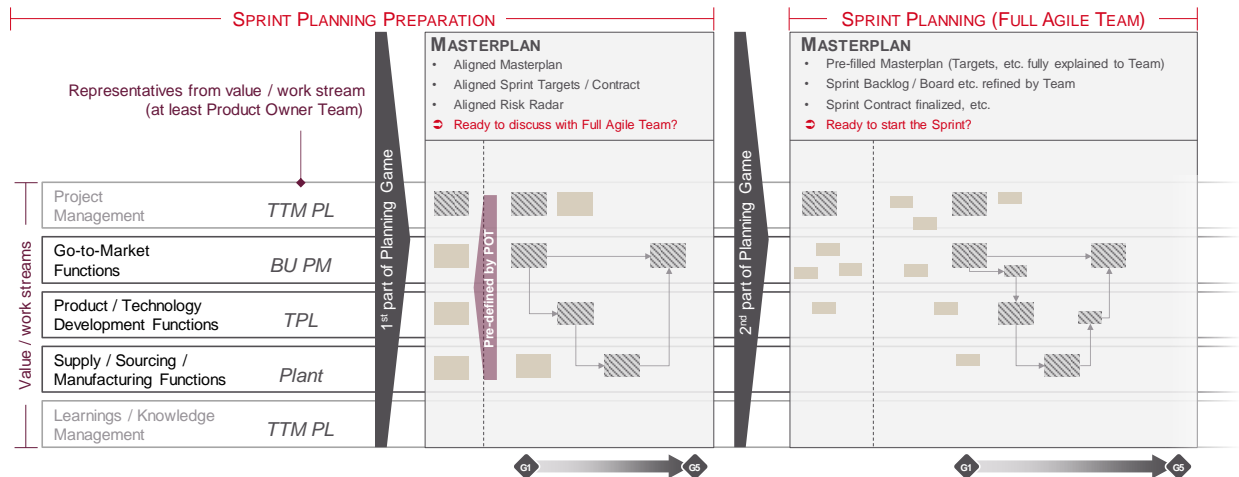


Figure 40. Planning Game in a nutshell © 2018 Hilti ALL RIGHTS RESERVED

The second part of the Planning Game starts with each team member to check their Targets listed on the Masterplan and then move the Targets from the Masterplan to the Sprint Board either by moving them directly or if the Target is of larger character, it might be necessary to slice it into several smaller Targets. It is up to each team member to estimate how many Targets he/she can achieve during the coming Sprint. Every team member then explains each of his/her Targets to the rest of the team. The most important Targets are then highlighted in the Sprint Contract as Must / Should deliverables. The Learnings that the team wants to focus on for the next Sprint are included in addition to the relevant risks associated with the content of the Sprint which in turn finalises the Sprint Contract.

The Sprint Planning is concluded with an assessment where each member makes a mark on a scale ranging from green to red, of how realistic he/she finds the Sprint goals. If the result sticks out by any one team member or by the whole group, a discussion is held. An overview with a “5W1H” approach is shown below.

– Planning Meeting –	
Question	Answer
Why Sprint Planning (Game)?	<ul style="list-style-type: none"> • Purpose is a joint commitment of a feasible Sprint Contract (what can be shown at the end of the Sprint) • Increase team spirit and foster self-reliable Target setting and effort estimation
What content?	<ul style="list-style-type: none"> • Team Onion Model aligned, capacity and resources available and committable • Detailed planning of the upcoming Sprint <ul style="list-style-type: none"> – Planned outcome of the Sprint – Risks that can be updated – Implemented learnings, etc. ↳ Definition of an overall Sprint Target (name for the Sprint indicating Targets [and/or risks]) • Any reasons that affect timeline (OTD) and problems meeting project and product requirements? • Consolidation of results to synchronise value / work streams for overall project
Who prepares and updates material?	<ul style="list-style-type: none"> • Agile Project Manager(s) to invite, joint preparation with POT • Pre-filled Masterplan and related materials as 1st part of Sprint Planning Game available before meeting • 2nd part to adjust Masterplan and Sprint Board (incl. dependencies) filled with feasible Targets • Risk Radar adjusted • Sprint Contract finalised and committed by all <ul style="list-style-type: none"> – Value / work stream Targets and overall Sprint Target – Targets to mitigate risks that might affect upcoming Sprint – Improvements to implement (from Learnings Board) • Team voting – only for the probability of reaching Sprint Contract / Targets
Who is invited?	<ul style="list-style-type: none"> • All Full Agile Team Members invited and must attend
When and how long?	<ul style="list-style-type: none"> • Once a month, second half of the “Full Agile Meeting” • Should take place in the project’s “Obeya” room • Depending on project complexity up to three hours
👍 Do’s	👎 Don’ts
<ul style="list-style-type: none"> • Be full time present at this meeting • Plan realistic Targets and your availability (Sprint capacity and vacation) 	<ul style="list-style-type: none"> • Do not hide potential issues regarding your work load / requested capacity.

The Sprint Planning is a very important Agile Meeting and thus covers at least half of the time of the Full Agile Meeting. Planning is key for the team to better estimate their own workload and potential Sprint outcome, but also for a better predictability of e.g. crucial cross-functional resources for future Sprints.

How much work does a team plan?

Since it is difficult to estimate the Targets effort jointly as a team with e.g. “Planning Poker” due to non-overlapping skillsets, it is simply up to the experts to estimate how much they plan

to achieve each Sprint by themselves based on their assumptions and previous knowledge about the topic.

"Daily" Agile Meeting

The daily meeting is only intended for information sharing for the most relevant people/functions in the current Sprint and the PO/POT, together forming the Core Agile Team. However, the members of the Core Agile Team can, if they find it necessary, extend the meeting for another 15 minutes if they feel the need to go more into depth in some topics. The members share information quickly and often by answering three simple questions shown below:

– Daily Meeting –	
Question	Answer
Why "Dailies"?	<ul style="list-style-type: none"> • Purpose is to exchange information among the whole team, esp. identify issues that affect team (members) • Increase transparency for the whole team
What content?	<ul style="list-style-type: none"> • Every team member is engaged to tell <ol style="list-style-type: none"> 1. What did I work on yesterday? 2. What will I do today? 3. What hinders me in my job? ➡ at least that is a must in case! (There might not be a need to answer questions 1. and 2. as in HW development things don't change overnight) • Identification of the need for further clarification
Who prepares and updates material?	<ul style="list-style-type: none"> • The Agile Project Manager(s) invite to the meeting, at least to a regular Skype conference call • The only must-update on the Masterplan might in case be the Decision Board
Who is invited?	<ul style="list-style-type: none"> • Invited are all team members • Attend must the Agile Core Team (POT, Agile Project Manager(s) and Core team members)
When and how long?	<ul style="list-style-type: none"> • Recommended is a daily exchange • Every member gets a maximum of 90 seconds to speak • Agile Project Manager in charge to cut speech
👉 Do's	👉 Don'ts
<ul style="list-style-type: none"> • Attend whenever possible • Directly get to the point, and announce need of further clarification in case of issues / hurdles 	<ul style="list-style-type: none"> • Do not get into technical or any other specific deep-dive discussion. • If deep-dive discussion is necessary, it is possible to extend the meeting.

The frequency of "Daily" meetings

The teams are recommended from the beginning of initiating the Agile methodology to have the Daily meetings once a day, suggested by the name. However, looking at the pilots, only Pilot Project B continued with Daily meetings once a day whereas Pilot Project A and C shifted towards only having the meeting twice per week arguing that since HW development has, by nature, a greater mixture of expertise needed from different functions and thus the skillset does not overlap as much as in SW development, it is unnecessary in their situation to

share information every day because it did not provide any additional value. Therefore, the team may choose how often they need to have the Daily Agile meeting. Pilot Project B continued with having the meeting once per day because the project's focus and characteristics were narrower.



Some teams found it challenging to stick to the proposed time constraint for the Daily Agile meeting, therefore the issue was raised and identified as a potential improvement.

One team member said:

The best is coming together as a team! Avoid problem in the project because of email collaboration. Emails very often can be misunderstood, they also never resemble the same communication as face to face. Also, time blocking for after discussion is perfect!

Extended Agile Meeting

The Extended Agile Meeting is a meeting initiated to synchronise team work with functional experts and to give the team an opportunity to discuss and make deep-dives into specific topics such as technical challenges and to share information with relevant stakeholders. The meeting invites value / work streams that are not assigned full-time to the project but still have a significant importance. In order to determine which value / work stream that is relevant it is necessary to know which phase the project has reached. For example, Packaging-function might not necessarily participate during a project's early phase but when the product reaches a certain maturity in its later phases, the need for that function to participate changes.

- Extended Meeting -	
Question	Answer
Why Extended Agile Meetings?	<ul style="list-style-type: none"> • Purpose is to frequently exchange project progress and discuss content with project stakeholders that are not in the Agile Core Team. • Include extended expert resources on a regular basis, so that they can better plan to commit / attend • Identify capacity, resource or any other conflicts in the various value / work streams
What content?	<ul style="list-style-type: none"> • The meeting starts with a "Daily" Agile Meeting as short information exchange • Deep-dives and specialised information exchange in the various value / work streams to support / adjust Target clarification and way forward • Consolidation of results to synchronise value / work streams for overall project
Who prepares and updates material?	<ul style="list-style-type: none"> • Agile Project Manager(s) to invite • Trend Charts (team voting) to be updated weekly, while discussed in a monthly Full Agile Team meeting • Burndown Chart(s) to be updated weekly, while discussed in monthly Full Agile Team meeting • Risk Radar (only new risks might be added to corresponding field) • Decision Board • Masterplan adjusted
Who is invited?	<ul style="list-style-type: none"> • Invited are all team members • Attend must the Agile Core Team plus Extended Agile Team members
When and how long?	<ul style="list-style-type: none"> • Recommended is a weekly exchange • Should take place in the project's "Obeya" room • Depending on project complexity up to four hours
 Do's	 Don'ts
<ul style="list-style-type: none"> • Prepare results / progress to be discussed in advance • In case of requesting decision makings, please provide potential alternatives 	<ul style="list-style-type: none"> • Do not forget to invite the specific resources / capacities needed for clarification of deep dive topics

Sprint Review

The Sprint Review serves as a look-back on what has been done in the Sprint and for the whole team to take part in the results and give feedback. The Sprint Contract is assessed to see if Targets were achieved, the risks had been reduced and how the learnings that were implemented worked out. Following, the Risk Radar is updated and the Masterplan controlled. If new requirements have been identified, these are made into Product Backlog items and prioritised according to their importance.



– Review Meeting –	
Question	Answer
Why Review Meeting?	<ul style="list-style-type: none"> • Purpose is to review the outcome of the Sprint jointly in the Full Agile Team (look-back) • The review is basis for the Sprint Planning
What content?	<ul style="list-style-type: none"> • Detailed review of the Sprint outcome, esp. the Sprint Contract <ul style="list-style-type: none"> – Product related – Risk that can be updated – Implemented learnings, etc. • If something tangible was created it needs to be brought to this meeting • Any reasons that affect timeline (OTD) and problems meeting project and product requirements? • Consolidation of results to synchronise value / work streams for overall project
Who prepares and updates material?	<ul style="list-style-type: none"> • Agile Project Manager(s) to invite, joint preparation with POT • Sprint Board (achieved Targets) • Masterplan (new Targets in Backlog) • Risk Radar (some “impact” changes? New risks and/or mitigations?) • Burndown Chart(s) • Trend Charts
Who is invited?	<ul style="list-style-type: none"> • All invited, all Full Agile Team Members must come • Additional Hilti Stakeholders welcome • (Lead) Customers welcome
When and how long?	<ul style="list-style-type: none"> • Once a month, recommendation to use the same day as “Extended Agile Meeting” (as review is included) • Is the first part of the “Full Agile Meeting” • Should take place in the project’s “Obeya” • Depending on project complexity up to three hours
👉 Do’s	👉 Don’ts
<ul style="list-style-type: none"> • Bring tangible increments to review if possible • Explain why a Target (from critical path given by POT) can be set to “resolved” from your expertise point of view • Celebrate successes 	<ul style="list-style-type: none"> • Do not stay away from discussions

Bringing results to the meeting

Team members appreciated when an increment of the product was brought to the meetings because seeing the results in real life added a dimension of understanding of the product and gave a good feel of how far the project had come. However, due to longer lead times in HW development, not every Sprint Review may include some tangible results in comparison to SW development.

Sprint Retrospective

The Sprint Retrospective serves to gain learnings from the previous Sprint and allows the team to reflect on what went well and what could be improved from a process perspective. During the Retrospective meeting, every team member assesses his/her satisfaction with the project and/or the process by putting votes on a scale. The result is then put in a Team Satisfaction Chart and tracked weekly.

Retrospective Meeting –	
Question	Answer
Why Retrospective Meeting?	<ul style="list-style-type: none">• Purpose is to review the Hilti Hybrid Methodology and thus continuously improving the process• It is a “learnings” discussion and thus key for transforming to a learning organisation
What content?	<ul style="list-style-type: none">• Reflect about what went well and what could be improved<ul style="list-style-type: none">– Meeting Culture / Collaboration in the process– Implemented learnings, etc.• Conduct a team voting (see also Extended Agile Meeting) and then discuss the trend charts in more detail• Spark ideation and creativity, e.g. with 2-by-2 learnings game to improve project and process• Most important: Team Satisfaction and action plan
Who prepares and updates material?	<ul style="list-style-type: none">• Agile Project Manager(s) to invite• Trend Charts (personal perceptions)• Learnings Board
Who is invited?	<ul style="list-style-type: none">• All invited, as it is part of Full Agile Meeting• No additional stakeholders welcome (what happens in the project from a personal perspective stays there)
When and how long?	<ul style="list-style-type: none">• Once a month, in Full Agile Meeting• Should take place in the project’s “Obeya” room• Depending on project complexity, one to two hours
 Do’s	 Don’ts
<ul style="list-style-type: none">• Speak up in case of not being happy with the way the project runs from a method and contribution point of view• Share improvement potentials in the team	<ul style="list-style-type: none">• Do not stay away from this meeting as it is not product maturity related. It is an important meeting to foster team satisfaction.

A key aspect of the methodology

Continuous improvement is key in the Agile Framework, therefore the Retrospective meeting is of great importance to surface things that went well or could be improved in either the process or project. The meeting enables a natural forum to discuss challenges and solve them. Team members agreed that having regular lookbacks prepared them for potential future threats that might have posed a different severity if identified later in the project.

5.2.5 Preparation for Project Execution

This chapter provides information regarding some practicalities of room bookings and the software that is recommended to be used for all tools and materials and execution of Targets / tasks.

Obeya

A dedicated project room, Obeya, is necessary which will contain all the material (including potential boards, charts and other necessary items) for the team to have a constant overview of the project and its progress. It is recommended to use the same room for all types of meetings (Daily, Weekly, Sprint Review/Retrospective/Planning) but if the project is big and have a lot of people and stakeholders involved and/or non co-located team members, it might be necessary to have several different rooms. The meeting days and time of the day should always be the same to make the meeting's culture less complicated.

Software Tools

The software that is recommended to be used for the Agile Methodology is RealTime Board (RTB) and the complementary software JIRA. Both tools are cloud based and assist with the entire chain of the methodology, from planning to doing and documentation. RTB gives a structured overview of the entire project with all the necessary charts, boards, Backlogs etc. JIRA acts as a personal task management tool for each individual team member to slice a Target into smaller and more detailed tasks, which are trackable on a Kanban Board (see Theoretical Framework).

- RealTime Board (RTB) for the Masterplan and associated agile materials on a team level
- JIRA to manage own Sprint delivery within a Kanban / Sprint Board on a personal level
- SharePoint for the documentation of project relevant material

RealTime Board

RealTime Board is a team collaboration software in Hilti Agile Hybrids. It is a powerful tool that can be adjusted in “real time” by any team member. RTB is highly customisable to the need of the teams, meaning that depending on what specific Agile Materials are being used, most of these can be included and visualised in the RTB interface, as can be seen in Figure 41.

What is unique with RTB is that the software is cloud based and all team members can edit and work with it simultaneously and track all the changes done including the possibility to see each team member's cursor. This opens for excellent collaboration possibilities between team members and stakeholders that are non co-located to access to the material 24/7..



Figure 41. RealTime Board interface

The benefits of using RTB was elaborated by one team member like the following:

Usually Real-time board makes it clear where we are. In other projects I have to ask the project manager and I receive an email back – I cannot see the big picture from an email. In Agile you can see everything straight away and you can also come back to it anytime you want, it's not dependent on people. So if the project manager is on holidays I can always log in to RealTime Board and I know what's going on.

JIRA

JIRA is a task and issue management solution which works as a Sprint / Kanban board that is made for each team member's personal task management. JIRA is highly integrated into Hilti's project eco system, such as Office, SharePoint integration and interfaces to RTB and vice versa. With the tool, it is possible to see all the value streams and each member's Targets and tasks.

With the cloud-based software JIRA, the team members can easily slice a large Target into smaller tasks so that it is easier to time-estimate and get a better understanding of the particular Target and how it will be achieved. In practice, a user chooses a Target from the Sprint Board, identify its building blocks (tasks) and works on those individually by moving the tasks from left, "To Do" through "In Progress", to "Review" and finally "Done". When all of the tasks are done, the Target is finalised. An example of JIRA in practice is shown in Figure 42.

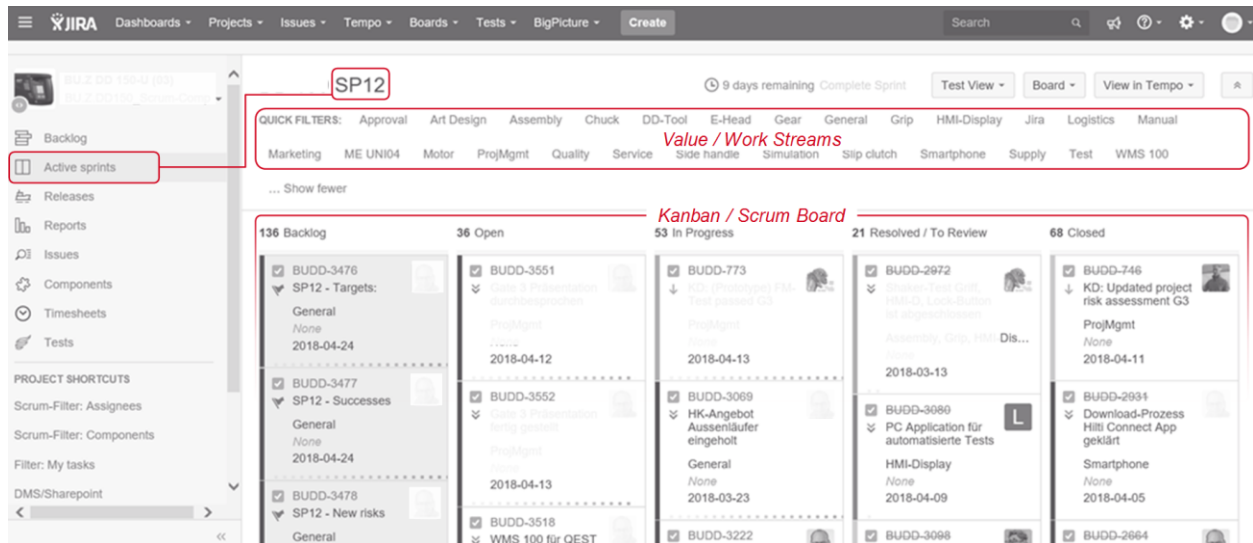


Figure 42. Personal Kanban Board in JIRA

One team member mentioned: *JIRA inspection provides transparency and great project visualization.*

Agile Coach

The Agile Coach is a temporary facilitating role that is fully engaged in a project whose members have previously not utilised the Agile Methodology. He/she is attending and assisting all meetings, events and planning. The coach provides training for the PO/POT and the whole team about the methodology, all the tools and materials, events, software, meeting culture and making sure that all the material (software and/or physical items) is available. The Agile Coach is important because he/she guides the team and motivates the members in case something in the process is not working the way it should or simply assists the individual team member to slice large Targets into smaller tasks to fit the Sprint. Figure 43 shows the Agile Coach's level of participation from the first Sprint to the fourth and how the team's need for support decreases as the PO/POT's confidence increases with time.

Sprint	External Agile Coach participation
1	Full facilitation
2	Facilitate together with TPL/TTM PL
3	Supporting TPL/TTM PL
4	Ad hoc support

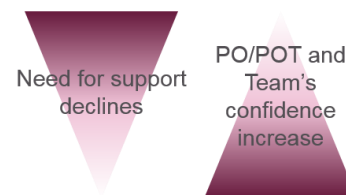


Figure 43. Agile Coach supporting

6 DISCUSSION, CONCLUSIONS AND RECOMMENDATIONS

This chapter includes reflections of the results and recommendations of potential changes to the process. Additionally, the Research Aims, stated Challenges, Problem Formulation and Research Question is discussed and assessed in order to see how well the study paired with initial statements.

6.1 Discussion and Reflections

The idea of using an Agile Framework in HW is generally bespoken as difficult and perhaps even redundant by many in the industry, however, the author's opinion is that the general perception may not be true to all cases. While it may not always be possible to use the exact same structure, tools, materials and meeting culture as in SW Agile development like Scrum, it is still possible to extract and adapt much of it, especially the mind-set, i.e. being agile in the sense of being ready for project changes and making big Targets / goals more tangible by slicing them into smaller Targets / tasks.

Sprint length

One of the more noticeable changes that had to be made was the strict usage of 4-week Sprint cycles because of the natural interdependencies between value / work streams and longer lead time of HW development. 2-week Sprints were tested but the team did not feel that it was enough time to deliver a viable result and having a Sprint meeting twice a month was also a bit too much giving the feel of being controlled rather than a conducting follow-up.

Methodology Freedom

The Hilti Agile Hybrid has a flexibility regarding how extensively the teams want to utilise the methodology's components. The teams can decide themselves depending on their own preferences and needs how much of the material and tools, meetings etc. that they want to use and make their own structure that suits the team best. A team that wants to adapt to the Agile Methodology is given knowledge on a conceptual basis which the team members can then adjust and tune according to how they prefer it. One of the most important aspects of the Agile Methodology is the continuous improvement like a self-learning organisation which is enabled when a team gets to experiment by itself and always tweaks and tries new ways to the approach. Like for example, when Pilot Project C used the Burndown Chart but ceased doing so when they realised that it did not provide the team any value.

Sticky Note (Target) Status

When it comes to the Sprint Board and Masterplan, the author recommends to use the same structure regarding the Targets (written on sticky notes) for all teams so that there would not be any confusion when/if a person would switch between two projects, which is very common.

Self-Learning Organisation

A good demonstration of how the Agile Hybrid Methodology enables teams to become self-learning is how they modify the tools used, like what Pilot Project B did with the Learnings and Successes. They implemented a column next to the Sprint Board which included both newly captured successes / improvements and some that were kept for more than one Sprint and served as a constant reminder of what the team has been doing well and/or may need to improve whenever the team members are adjusting the Sprint Targets.

Controversial Trend Charts

The usage of the Risk Trend tracking is controversial because there is no fixed highest level of risk or some sort of reference point. The trend chart does not measure a percentage of Burndown but rather in absolute numbers making the benefit of tracking the risk trend controversial. It is impossible to assume that all the risks have been identified from the beginning and that the total Risk Impact Value is being cut down methodically throughout the project. The team members' opinion about the radar is nevertheless positive because it still gives a hindsight of how the total risk of the project changes with time. Therefore, the usage of the risk tracking is highly dependent on whether or not the team feels that it provides any value. The author would recommend to find a way to incorporate more substance to the measurement and make it more tangible so that Agile teams can benefit more from the tool.

Decision Documentation

Documentation of decisions by using a decision board is a tool that enables any team member, no matter if he/she was present during the decision making, to view what changes / adjustments have been made. However, for the teams that use a physical decision board, it is recommended that a smooth online documentation tool is used which includes all decision at one place instead of a picture taken of the board each time a new decision was made and uploaded to a folder, which was the strategy of Pilot Project B. Microsoft Excel or OneNote could be potential options.

Target Estimation

To improve the planning capability for Targets of the team for each Sprint, and for the whole team to understand the effort needed for each other's Target to be achieved, a structured way of giving the Targets an effort estimation is needed. If the team members are assumed to work 40 hours per week, they could use a point-based estimation with a maximum points given per week 40 if each point is worth 1 hour of work. In addition, to provide further correct estimations and for the knowledge to be reused in future projects and with other team members, the author recommends that the estimated effort (points) *and* the actual effort is documented to be able to give more accurate estimations for similar tasks / Targets in the future.

Conflict of Interest

In other Agile Frameworks, like Scrum, the Product Owner has the role of pushing the team and essentially bring what needs to be done in the project to the table while the Agile Project Manager (a.k.a. Scrum Master in Scrum) is supposed to be facilitating the Agile process and ensure that the planning is reasonable and that each team member does not plan more than he/she can muster. Therefore, there is a healthy conflict of interest between the two. However, in the Hilti Agile Framework, the Agile Project Manager and one of the Product Owners can be the same person which might leave the team members and their personal effort overshadowed by the progress of the project if the person in charge does not consider his/her ambiguous role. Therefore, it is recommended that a PO is not the same person as the Agile Project Manager.

The Extent of Software vs Physical Tool Utilisation

The usage of software is an option for the teams. In many cases, there is a clear benefit to use facilitating software such as RTB and/or JIRA, especially for non co-located teams. JIRA also provides complete transparency by enabling every team member to see each other's work at

any time by just signing in to the software. However, if a team is 80-100 % co-located, a complete paper-based process solution might be a better option if the team prefers it that way. Several people from the Pilot Projects emphasised the importance of moving and feeling physical tools and materials, like writing on a sticky note and putting it on the board. It made the Targets and tasks more tangible and gave a feeling of working together and interact in a team effort. The Author believes that a middle course is the best way, i.e. using both physical and virtual tools combined to an extent that suits the team best. For kick-offs and the first couple of Sprints there is a value in keeping physical tools and materials for the team members to properly learn the methodology in a more interactive way and then transform the project into a virtual environment. But if the team since previously is already familiar with the methodology, using the virtual environment might work just as well.

Enhanced Company-wide Resource Allocation

Planning of resource allocation and timed capacity can be a struggle in companies. If the methodology is rolled out globally which means that most teams use the same or a very similar process, it might be easier to plan projects because of a constant rhythm of 4-week Sprints. The Agile Framework easily depicts each person's commitment to his/her projects and if everyone plan their commitment on a Sprint basis, aligning work may get easier. The benefits of the Agile Framework for resource allocation can be highlighted by some quotes from team members.

One Project Manager mentioned:

Agile techniques are by far the most efficient ones in terms of project management! I know all of the team members, I know exactly what they do and when they do it. Very often this is the struggle for me in any other project. Finding who is doing what is extremely important.

Another said:

We have a good common overview of what's involved in the project. We can see the big picture and this includes all of us. We have a common ground to discuss things to work out our aims and direction as a team.

Customised Meeting Culture

The meeting culture is perceived by some team members as overwhelming with for example a meeting every day. It is however independently set by each team how many meetings they find reasonable to have, and how often. The Hilti Hybrid Framework gives the recommendation by default to have the "daily" meeting every day but in a less complex project, it might not be entirely needed to exchange information that often. It is important to not let the meetings become a controlling session. This could leave the team members with a feeling of constantly being monitored and thus limit the sense of freedom and self-responsibility. The methodology is thought to empower the individual team member, instil trust and give support to each other instead of questioning and controlling each other's work.

Differentiate Methodologies

The Hilti Agile Hybrid Framework is an entirely new process that shares some similarities with Scrum. To avoid mixing up the methodologies, the naming of tools, materials, meetings etc. have gotten a unique character. For example, in Scrum, "Agile Tools and Materials" are called "Artefacts" and the "Daily Agile Meetings" are called "Scrum Meeting" in Scrum. The content is essentially the same but the naming should not liberally be exchanged in order to be able to distinguish the methods. The reasoning is partly that people who have previous experiences with Scrum might therefore have mixed pre-determined thoughts of how the

methodology is executed and what might be good or not with the content of the Framework. The most important aspect is to have a logical distinction between the new methodology and other methodologies.

6.2 Research Aims

The study was initiated with stating aims that the research anticipated to reach. They were:

- A documented concept utilising Agile/Scrum in hardware and complex systems development which integrates with current Phase-Gate methods
 - The research resulted in a handy guideline of how to utilise the methodology in a practical manner including analyses and reflections. The guideline can be used by teams that have the ambition to initiate a project that wants to apply the Hilti Agile Hybrid Methodology. The method uses a combination of the Phase-Gate and Agile Frameworks. Therefore, this aim is considered reached by the author.
- A contribution to the definition of the business unit
 - With the current future horizon of the company, the methodology is anticipated to become a part of the portfolio of Hilti's processes. The result of the research provided a step towards a higher maturity of the process to be rolled-out in the future. A contribution to the business unit Product Portfolio Management can thus be concluded.

6.3 Challenges and Problem Formulation Revisited

With the background of what was stated in the Problem Formulation about what challenges the research anticipated to address, how well were these achieved?

- Interdependencies between internal (activities in engineering and manufacturing) and external (suppliers, integration, approvals, etc.) functions
 - The interdependencies are addressed by using a Masterplan and Sprint Board to visualise the critical path between Targets. In addition, certain meetings invite and thus enable the different functions or value streams (internal and external) to discuss and resolve potential challenges on a regular basis.
- Cross-functional and cross-organisational project contribution
 - With the Team Onion Model it is possible to easily get an overview of the meeting culture and thus knowing who meet when and where. Depending on the relevancy of the value / work stream, different contribution from cross-functional and cross-organisational is needed
- When global Project Teams are not co-located
 - Teams can use the live-updated software RealTime Board and JIRA to assist the project execution. Since they are cloud based it is possible to access the material anywhere at any time, given that there is an internet connection. This lets teams from all over the world more easily collaborate in projects, especially those which use the Agile Framework.

- Making the final stakeholder a part of the team
 - According to the Hilti Agile Framework, the team can include the final stakeholders in a joint team meeting once every Sprint to view the progress and assure that it is moving in the direction that the stakeholders want it to. This was not done as often with the previous methodology which may result in a product which is not entirely in line with the stakeholder's expectation.

6.4 Research Question

The *Research Question* was formulated as the following:

How can Hilti adapt and integrate an Agile Methodology with their Phase-Gate Methodology in Hardware Development?

By conducting a literature review, fact research and exhaustive consultation with teams throughout the study, the author made detailed documentation with best practices including self-reflections and improvement proposals which was eventually concluded in a thorough walk-through of the entire development process, depicted in Chapter 5, The Hilti Agile Hybrid. This documentation /guideline has the inherited flexibility, discussed in Chapter 6.1 Discussion, making it possible to adjust projects with varying characteristics. In that sense, it is reasonable to conclude that the *Expected Outcome*, formulated like the following:

A comprehensive documentation with step-wise instructions, applicable and adapted to fit multiple categories of Hilti HW Development

is in line with the result of the research.

6.5 Future Work

The target workload estimation is proven to be difficult since there is only one or a very limited number of people who have enough knowledge in each separate area who can make a proper workload estimation. Scrum in software development has a more accurate workload estimation because there are more people who can assist and give input to the various target workload estimations. A future work could be to develop a better concept of how to conduct more accurate estimations by for example inviting experts from outside the project who can provide proper input.

The methodology is infinitely adjustable to fit basically any type of project. For example, a project of a research character could probably be within scope by adapting the meeting culture more appropriately and perhaps make sprints longer than four weeks since research is often time-intensive. This leads to a potential future work where more specified concepts, customised for particular project types is developed. Implications would include adjusting the definition of some Agile materials and tools and meeting contents to better fit the project type in question.

7 REFERENCES

- Agile Alliance, 2001. *Agile Alliance*. [Online]
Available at: <https://www.agilealliance.org/agile101/the-agile-manifesto/>
[Accessed 19 01 2018].
- Agile Alliance, 2018. *agilealliance*. [Online]
Available at: [https://www.agilealliance.org/glossary/scrum-of-scrums/#q=~\(filters~\(postType~\(~'page~'post~'aa book~'aa event session~'aa experience_report~'aa glossary~'aa research paper~'aa video\)~tags~\(~'scrum*20of*20scrums\)\)~searchTerm~::~sort~false~sortDirection~'as](https://www.agilealliance.org/glossary/scrum-of-scrums/#q=~(filters~(postType~(~'page~'post~'aa book~'aa event session~'aa experience_report~'aa glossary~'aa research paper~'aa video)~tags~(~'scrum*20of*20scrums))~searchTerm~::~sort~false~sortDirection~'as)
[Accessed 13 02 2018].
- Anon., 2018. *Wikipedia*. [Online]
Available at: https://en.wikipedia.org/wiki/Phase-gate_process
[Använd 03 02 2018].
- Beck, K. et al., 2001. *Agilemanifesto*. [Online]
Available at: <http://agilemanifesto.org/>
[Accessed 03 02 2018].
- Cooper, R. G., 2008. Perspective: The Stage-Gate® Idea-to-Launch Process—Update, What's New, and NexGen Systems. *Product Innovation Management*, 25(3), pp. 213-232.
- Don, K., 2016. *Workfront*. [Online]
Available at: <https://resources.workfront.com/project-management-blog/project-management-101-the-5-qs-and-1-h-that-should-be-asked-of-every-project>
[Accessed 30 4 2018].
- Edgett, S. J., 2018. *Stage-gate*. [Online]
Available at: https://www.stage-gate.com/resources_stage-gate_lessonslearned.php
[Använd 03 02 2018].
- Hilti, 2018. *Hilti*. [Online]
Available at: <https://www.hilti.group/content/hilti/CP/XX/en/company/corporate-information/company-profile/company-profile.html>
[Accessed 19 01 2018].
- Kniberg, H. & Skarin, M., 2010. *Kanban and Scrum - making the most of both*. s.l.:C4Media Inc..
- Leopold, K. & Kaltenecker, S., 2015. *Kanban Change Leadership: Creating a Culture of Continuous Improvement*. s.l.:John Wiley & Sons. Books24x7.
- Mahnič, V. & Hovelja, T., 2012. On using planning poker for estimating user stories. *The Journal of Systems and Software*, p. 2086– 2095.
- Maximini, D., 2015. *The Scrum Culture*. Wendlingen: Springer.
- Mynott, C., 2012. *Lean Product Development*. u.o.:The Institute of Engineering and Technology.
- Ocamb, S., 2013. *Scrumalliance*. [Online]
Available at: <https://www.scrumalliance.org/community/articles/2013/2013-april/what-does-the-agile-manifesto-mean>
[Accessed 03 02 2018].

Reynisdóttir, Þ., 2013. *Scrum in Mechanical Product Development*, Gothenburg: Chalmers University of Technology.

Scrum Alliance, 2001. *Scrumalliance*. [Online]

Available at: <https://www.scrumalliance.org/why-scrum/core-scrum-values-roles>
[Accessed 05 02 2018].

Scrum.org, 2016. *Scrum*. [Online]

Available at: <https://www.scrum.org/resources/what-is-scrum>
[Accessed 04 02 2018].

Scrumguides, 2017. *Scrumguides*. [Online]

Available at: <http://www.scrumguides.org/scrum-guide.html#events-sprint>
[Accessed 11 02 2018].

Sims, C. & Johnson, H. L., 2012. *Scrum: A Breathtakingly Brief and Agile Introduction*. s.l.:Dymaxicon.

Takeuchi, H. & Nonaka, I., 1986. The New New Product Development Game. *Harvard Business Review*.

Testingexcellence, 2017. *Testingexcellence*. [Online]

Available at: <https://www.testingexcellence.com/common-agile-development-methodologies/>
[Accessed 22 01 2018].

UMSL, 2009. *UMSL*. [Online]

Available at: <http://www.umsl.edu/~hugheyd/is6840/waterfall.html>
[Accessed 29 01 2018].

Wikipedia, 2018. [Online]

Available at: https://en.wikipedia.org/wiki/Waterfall_model
[Accessed 19 1 2018].