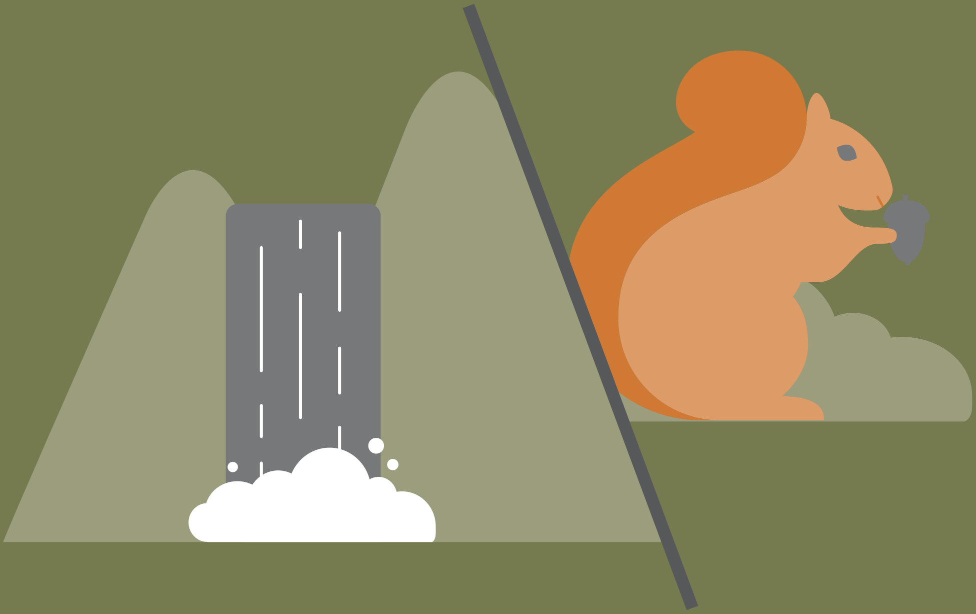
AGILITY AND PROJECT MANAGEMENT



kaastrup | andersen

Agility and project management

kaastrup|andersen has vast experience with both traditional and agile approaches to project management - and the hybrid models that are customized to fit the individual needs of the organization. We have written a number of articles, providing you with insights and overviews.

Enjoy!

Contents

INTRODUCTION TO **AGILE PROJECT MANAGEMENT**

Project management in transformation

Today, many businesses face practically ruthless demands to be ready for changes due to the constantly shifting needs of their customers. On top of that, technological developments often overtake companies from within, and changes to market conditions come fast and with little warning.

This also means that project management is undergoing transformation. Agile methods are gaining more and more ground based on the assumption that projects are not static, and that new knowledge can (and should be) be acquired along the way and applied to the project immediately. In short, agile project management has become the solution to some of the weaknesses that plan-driven project management has struggled with for decades.

What are we talking about?

The word agility derives from the Latin *agilis*, which means *easily moved*. Other English words include *versatile, nimble, flexible, quick, spry.* If we think further, even more words pop up: *adaptive, organic, cooperative, changeable, open to alterations.*

Agility as a way of thinking is one thing, and another is the concrete IT development methodologies characterised by agility. In other words, we can talk about something mental: a kind of “mindset” or “culture”, and a tangible set of tools that support this agile mindset.

The agile world consists of many philosophies and methods, which in time have become practical tools and frameworks (such as Scrum, Kanban, SAFe, PRINCE2 Agile), each of which presents its own suggestions for managing different parts of the agile process. The various elements of the agile world make it difficult to clearly identify what best suits one’s own business.

Why do we want to be agile?

A major study conducted in 2017 describes the goals a large number of businesses have for taking the agile path:

75%	64%	55%	49%	46%
Deliver products faster	Handle continuous changes better	Increase productivity	Facilitate better compliance between the business and IT	Improve the quality of software products

Source: www.collab.net

In several agile frameworks, the project manager role is said not to exist, with autonomous groups taking their place. You could therefore challenge our title, “Introduction to agile project management”. However, our experience shows that the need for project management does not disappear in an agile set-up. Instead, the role takes a different name, and its contents are different as well. Read more in the article “*Where is the project manager in an agile set-up?*”.

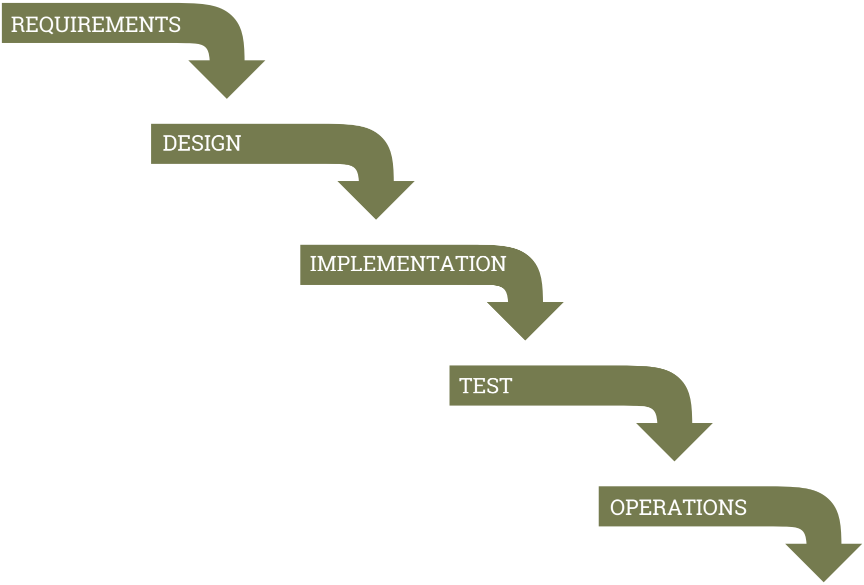
Agile methods – the short version

An agile method such as Scrum features a more incremental and iterative approach than the plan-driven and sequential methods we have traditionally used in our work. Instead of intensive planning and design in advance, agile methods allow for ongoing changes to product requirements. Teams working on the project include planners, designers, developers and testers, who work on iterations of a product through “locked” time periods (known as *Sprints* in Scrum).

Work is organised through a Backlog (see the glossary for a definition), which includes a prioritised task list from the business. The teams are self-organising, and also include representatives from the business (such as in the Scrum role *Product Owner*). The focus is on efficient face-to-face communication and short feedback loops. The goal of each iteration is to produce a functioning product to be used by the stakeholders. Feedback can thus be implemented in the next and upcoming iterations.

PROJECTS: **TRADITIONAL OR AGILE APPROACH?**

When we talk about traditional project management, we refer to plan-driven project management. The *waterfall model* is often used as an example of the traditional, plan-driven way of handling projects and larger tasks. The *waterfall model* is a sequential, phase-driven model, where the next phase (in theory) does not begin until the previous one is completed. One major characteristic of the traditional approach is that all requirements are stipulated in detail before the project begins, so that the project can be managed according to the requirement specifications. The focus of the project is to meet the requirement specifications. Changes can be a large and heavy burden, as requests for changes from users tend to appear late in the timeline.



REQUIREMENTS

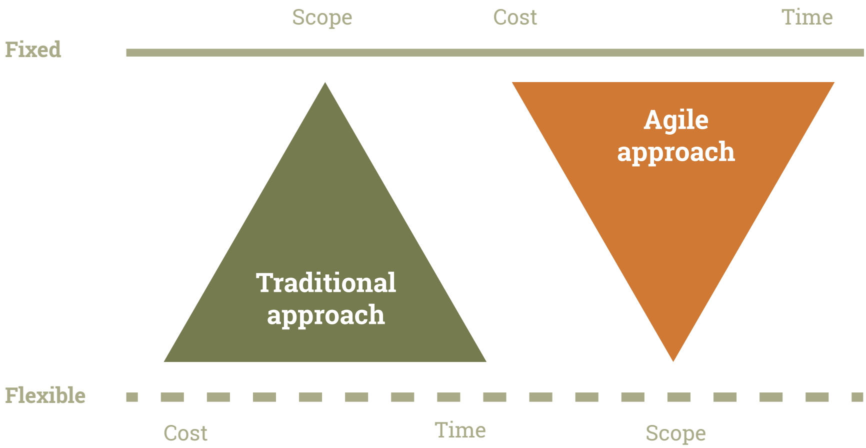
DESIGN

IMPLEMENTATION

TEST

OPERATIONS

In the agile approach, you make several iterations throughout the phases of the project. The customer is actively involved already at the start of the project, and priorities as well as overall goals are known to everyone working on the project. The project provides small bits of functionality along the way, allowing the customer to quickly make use of the product. Detailed requirements are made along the way in collaboration with the customer as the final project gradually takes form, so the focus can be maintained on the user's needs. One could describe the approach as change-driven.



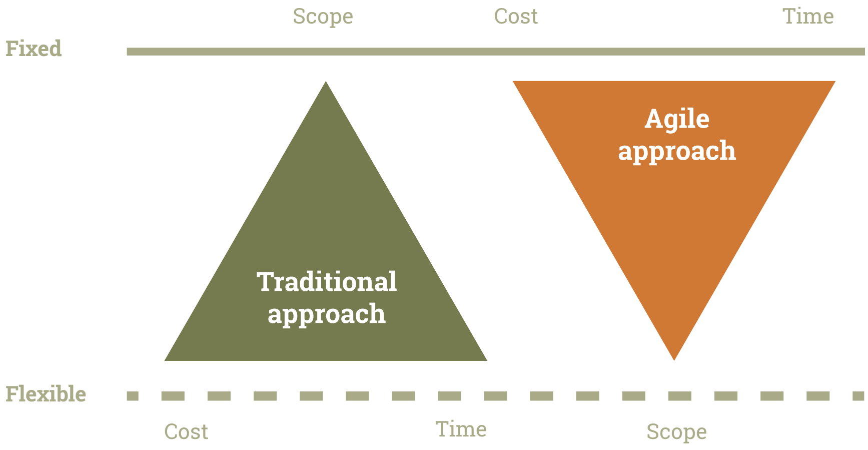
With the agile approach, the aim is to confront static, plan-driven projects, which in many places have a reputation for exceeding budgets, taking longer than estimated, and ultimately yielding less value to the business than expected.

One of the biggest concerns for many managers in transforming an organisation from plan-driven to agile management is the uncertainty as to whether one is able to maintain an overview of the scope and costs. Many fear that agile projects may go on forever. In principle, we do not refer to an agile "project", but we say that products in an agile reality are in development and optimisation at all times (all depending on the needs of the business). However, there is nothing preventing an agreement that certain efforts relating to a product will have a specific duration.

Project foundation

The traditional contract (the project brief) for a project has its foundation in estimates based on scope, time available and costs (as well as available knowledge). These are elements that may be difficult to take into account beforehand, but which will control the run of the project.

A contract that is to support an agile project sets costs and time available in the beginning, while scope is decided in cooperation with the customer and supplier during the course of the project. The requirement specification is replaced by an estimated backlog, which sets the framework for what can be included in the scope. The most notable difference, looking at the overall framework of the project, is that the scope is flexible in the agile approach. The final product depends directly on the customer's input along the way, and can end up looking completely different than expected at the beginning of the project.



How are these differences seen in practice?

This concrete example of the two different approaches bases itself on a scenario involving application (app) development. The two approaches handle tasks differently:

The traditional method starts by creating project documentation in the form of a project brief and an exact requirement specification, producing a detailed project plan, confirming project groups and management groups, and getting documents and decisions approved before finally beginning development. The whole process can take a long time.

If the customer (or end-users) have changes to make to the product requirements during the run of the project, the traditional method will require the project manager to handle an change process, which typically consists of many steps relating to re-estimation, impact and cost assessments – and then adjust the requirement specifications, test scenarios and project plans. All changes must furthermore be approved by the steering committee and the project owner.

The agile approach focuses on finding a Product Owner in the business, who (in cooperation with a customer or users where relevant), defines the overall vision for the app, in the form of a description of the needs it has to meet. A Scrum Team must also be gathered, and a Scrum Master is appointed before development can begin.

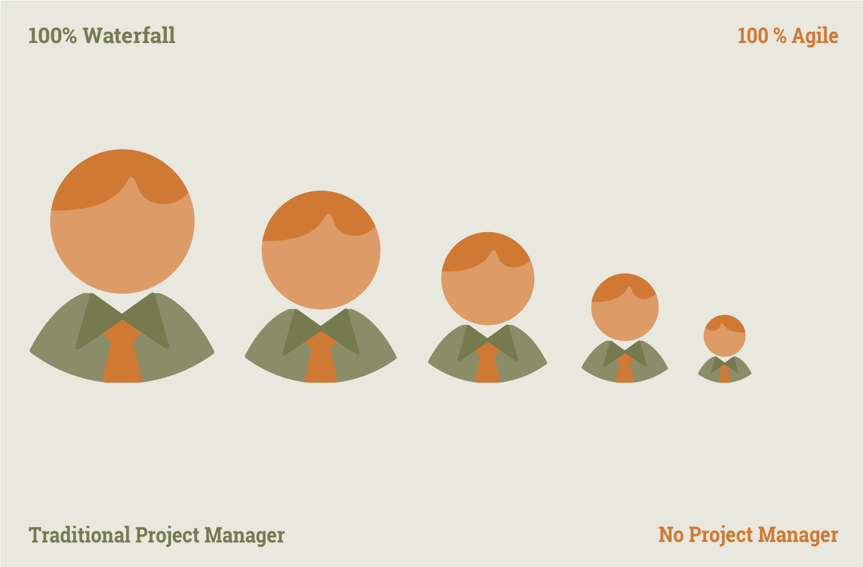
A lot of time is saved in the starting phase of the agile method, which in return demands that both the Product Owner and customer:

1. have the required decision-making competence and
2. are willing to (this applies in particular to the customer) spend time and energy to participate in development efforts to ensure the relevance of the product.

In the agile world, continuous changes are a part of the process. The customer (or end-users) help continually test and evaluate all deliverables and can adjust the direction of the development along the way. The overall vision for the product, which was described at the beginning of the project term, says something about the needs the product is intended to meet, and the form and functionality of the product is determined along the way. In practice, adjustments are made continually in an agile project.

WHERE IS THE PROJECT MANAGER IN AN AGILE SET-UP?

Where is the project manager in an agile set-up?
When we describe the project manager role, we often base our view on the traditional project manager role in the waterfall model. In an agile set-up, there is no role called “project manager”, but reality shows that many of the tasks a traditional project manager performs do not immediately vanish as most organisations only implement partial agility.



In eality, only few organisations are truly either 100% one or the other. Far more are somewhere between those two extremes. This leads to a wide range of challenges and opportunities for both organisations and project managers, and calls for reflection, cooperation and competence development.

In the following texts, we take a closer look at both extremes and the middle of the scale, and present some suggestions on how the project manager role can be transformed as the organisation finds its new place in an agile set-up. We choose to use the word "project manager" while remaining well aware that the role does not exist in a 100% agile set-up. There are however only few organisations that are 100% agile, so in reality, the word "project manager" continues to be useful.

Theory states…

If we take an overall view of the two extremes, several factors set the two approaches apart. Project managers that have to go through both environments will discover that the further they move from the waterfall-oriented world to the agile world, the clearer the paradigm shift is.

Fundamentally, the project manager role in the agile world will place considerably higher demands on the project manager's abilities as a manager, as the manager in the agile world must **lead** rather than **manage** people to a greater degree.

The core competencies for a manager in an agile set-up are:

- being able to create vision and motivation
- being able to set a direction based on competencies in business and strategic management
- being able to serve as a coach for, and shield the team
- being able to get involved and support the human resources in the project

In short: It is more about co-leadership than leadership by authority.

A traditional project manager who seeks to work agile will furthermore be challenged on his/her use of traditional management elements. The agile world challenges the traditional project manager role's assumptions on value-setting and norms, such as with regard to how a development timeline should be planned, managed and documented. This means that management and leadership must play out in a different way than before:

- In the agile world, needs appear continuously as a result of the process, while requirements in the traditional world are decided and described in advance.

- In the agile world, changes are necessary and welcome, in the traditional world they are often a seed that gives rise to re-planning, management group meetings, and so on.

- In the agile world, development (such as for a new product) is change-oriented, incremental and iterative, while development in the traditional world is plan-driven, sequential and determined in advance.

- In the agile world, progress is measured by the volume and quality of the delivered product, while progress in the traditional world is measured by the consumption of resources, time, money, etc.

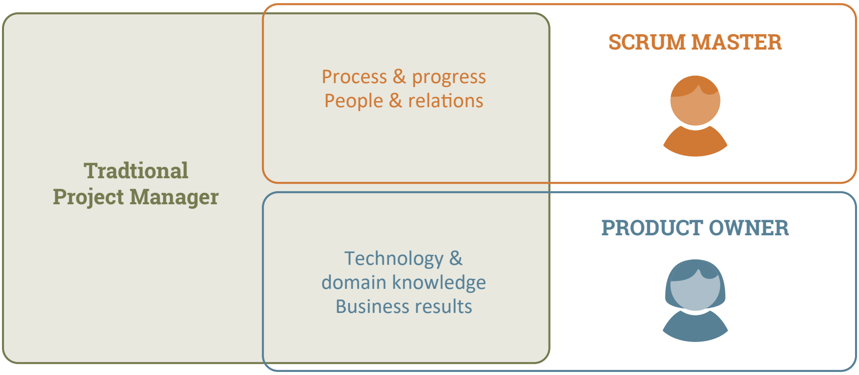
- In the agile world, the focus is on delivering a usable product after each iteration, while the focus in the traditional world is on following fixed rules and procedures.

And in reality?

Seen from this theoretical perspective, where the lines are sharply drawn, there are many things a traditional project manager must address and change in their management style if he/she wishes to go from using waterfall methodologies to using the agile method.

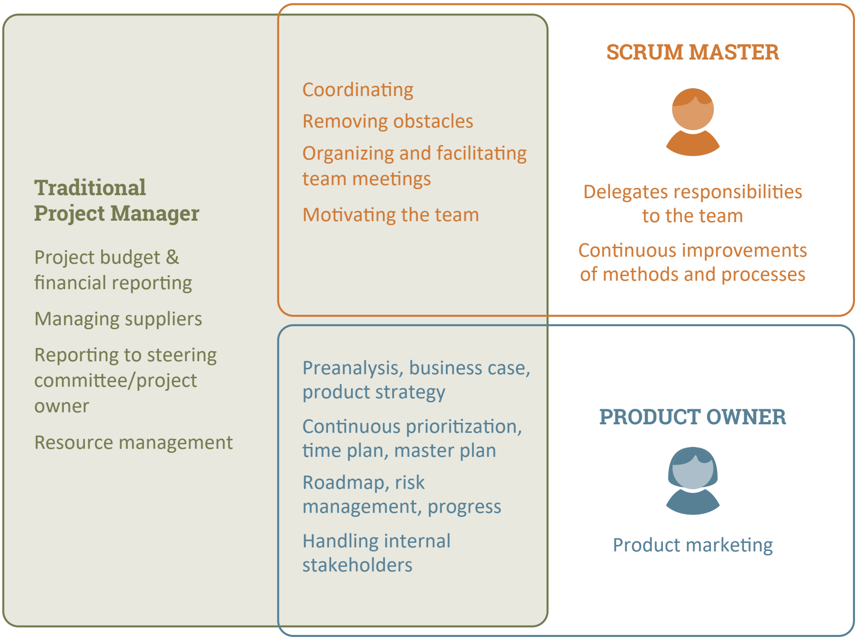
In practice, in the real world, we do see that most project managers have already begun the transformation, and are working from a mind and value set that is close to the agile. The best project managers know the difference and are able to balance their use of competencies and tools from both worlds, tailoring their methodology to suit the given project in the given organisation.

Scrum is the most commonly used agile method. If we take Scrum as an example, we can paint an overall picture of where the traditional project manager roles land: in Scrum Master and Product Owner (see the glossary for a closer description).



The figure shows how tasks and responsibilities are distributed in an organisation that works 100% agile. There are however very few organisations located at 100% in the "agility scale". Many are undergoing a transformation, some aim to reach 100% agility and others wish to let the results determine how much agility they implement in the long term.

The fact is therefore that by far the most organisations are somewhere between traditional and agile methodologies. At the same time, there is a range of tasks that are not included in the above figure. On the next page is a suggestion for a more nuanced picture of the reality, detailing the split of the tasks related to a project. This can be used to assign tasks and responsibilities:



Traditional Project Manager

Project budget & financial reporting

Managing suppliers

Reporting to steering committee/project owner

Resource management

Coordinating

Removing obstacles

Organizing and facilitating team meetings

Motivating the team

SCRUM MASTER

Delegates responsibilities to the team

Continuous improvements of methods and processes

Preanalysis, business case, product strategy

Continuous prioritization, time plan, master plan

Roadmap, risk management, progress

Handling internal stakeholders

PRODUCT OWNER

Product marketing

Again, a good project manager is one who is able to ensure an appropriate coverage of these management tasks through dialogue and expectation matching with the Scrum Master and Product Owner. In the vast majority of examples, dialogue and expectation matching with the surrounding organisation is necessary as well: Often, surrounding parties lack clarity about what agile methods entail, or how to get the most value out of them. The project manager must understand how the organisation works and use the relevant tools from the toolbox – both traditional and agile.

Advantages to using the distribution above

For organisations that are not 100% agile, this is a good approach to cover all relevant management aspects when running a project and delivering value to the business. The approach accommodates the agile elements while including the management, leadership and reporting elements in traditional waterfall models.

Disadvantages to using the distribution

If the organisation is not ready and willing to operate with agility, then some challenges may arise. Mixing the approaches demands a deep, shared understanding among all stakeholders regarding the ground rules you are following. Similarly, the difference in the way waterfall and agile methods handle the project triangle may be a dilemma that makes management tasks more difficult.

HOW DO WE TRANSFORM THE PROJECT MANAGER – AND THE ORGANISATION?

Many of the elements that are not included in product delivery issues, such as budgets, status reporting, management group handling, portfolio management, release management and vendor management will not disappear when agile is used. In the traditional world, these tasks are part of the project manager’s role, but in an agile set-up, they are distributed across more roles. The transformation to being a project manager in the agile world is about relaunching the term “Project Manager”, where the function may appear across several roles.

1. Educate and train existing project managers in the agile mindset based on holistic management considerations. A Scrum Master course alone will not transform a traditional project manager to an agile project manager.

2. Educate existing project managers in the most important roles that are present in an agile organisation, so that the project manager knows the theoretical foundation for the areas of responsibility and interactions shared between the agile roles.

3. Take an active stance on which levels the business should practice agile management in. A clear picture must be made of the agile organisational structure that is to be used for both the project manager and the organisation. The various levels could for example be: Portfolio management, Project management, Vendor management, Product development.

4. Educate and train project managers in the frameworks and methodologies that are to be used in the business. This could include Scrum or Kanban.

5. Develop an agile project model for the business with defined roles and areas of responsibility. Note that there may be a need for different models for different types of projects: Agile software development requires a different model than agile ERP implementation. The key point is that the project manager needs to have a deep understanding of the selected project model(s).

7 TIPS FOR PROJECT MANAGERS IN AN AGILE CONTEXT

Coordinate expectations!
The Scrum Master, Development Team and Product Owner must share an understanding. Just as in traditional projects, you must not forget to make a structure that works (and can change continuously, if necessary), and to agree on roles and responsibilities. It is also important to thoroughly customise the agile process, so that it suits the context the project operates in.

Ensure communication!
As opposed to traditional project approaches, where the focus is on plans and deadlines, the agile approach focuses on scope, which is more loosely defined and changeable. The chance of the scope changing along the way makes it important to communicate and coordinate expectations continuously and thoroughly with relevant stakeholders.

Get full-time resources!
If at all possible, make sure to get full-time resources on the project. The alternative is to be precise on what the team can achieve, so that the contents of each Sprint suit the efforts that can be made. The purpose, speed and capacity of the team (how much can be delivered per Sprint) must be assessed at all times, so that it can form the foundation for a continuous coordination of expectations with relevant line managers.

Optimise team cooperation!
Focus on the composition of the Development Team: What is needed to create and support a stable, effective, well-functioning and cohesive team? There may be challenges where the team is split geographically or comprises people from many different cultures. Focus on preventing and removing internal conflicts.

Remember: It's OK to fail!

The expression “fail fast” makes a lot of sense in an agile world: It is about being willing to let deliverables undergo tests and accept that they may fail. The approach ensures that no unnecessary time is spent developing a product in a direction that later proves to be misguided or unproductive. The retrospective meetings after each Sprint are to ensure that the experiences drawn from the failures are used to improve upcoming Sprints. Remember that the purpose of this attitude to failure must be communicated and coordinated with the stakeholders.

Clarify the responsibility for the backlog!

The Product Owner is responsible for the backlog and its prioritisation, but other people in the team can also take responsibility. And remember, there can be only **one** priority 1! Focus on making sure the right people take part in describing each task in depth, before the task becomes a part of the Sprint Planning.

Adapt your planning!

Consider how much planning makes sense in relation to each Sprint. In one concrete example, a team used to start with about 1 day for 8 developers to plan a Sprint of 10 working days. After some Sprints, 8 developers now spend about 2 hours to achieve the same result. Practice makes perfect!

AGILE PROJECT MANAGEMENT – DOES IT MAKE SENSE FOR YOU?

When should you consider using agile project management in your organisation?

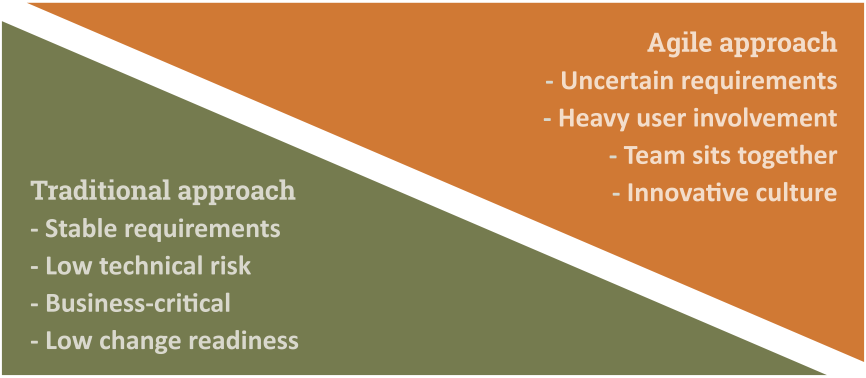
Agile methods and philosophies can be used in all types of projects across various industries - both in product development and IT software development. The wide perspective means that agile methods and approaches are ideal for all projects where a high level of innovation and user involvement is desired. The precondition for agile project management to be successful is that you have an organisational culture characterised by trust, open communication, transparency and the employees' ability to organise themselves in relation to the tasks that need to be completed.

For many businesses directing their attention towards agile methods, the decisive factor will be the degree of agility they are ready to implement at different levels of their organisation. At a completely fundamental level, you challenge your organisation on the extent of its management and control, and there may be geographical and cultural challenges as well. When it comes to projects in environments with a high degree of complexity, marked by considerable uncertainty, perhaps handled by an unstable organisation with regard to teams and more, then the agile journey may be a bigger challenge than when the opposite is the case.

Agile methods work best when management clearly supports them, as it is about creating the shortest path from decision to execution. This means delegating leadership and responsibilities downward in the organisation to business managers and executing teams. This also brings the implementation of new values in the organisation, as well as changes to the business culture. The implementation of agile methods may therefore have wide-ranging consequences for a business and the characteristics of a paradigm shift for the company.

There is no universal solution

The below figure can be used when deciding whether to use agile methods or to hold onto a more traditional, plan-driven approach to product development. The deciding parameter is often “uncertainty”. If enough (or all) of the issues that are to be solved are known and are most likely not going to change before the completion of the project, then a plan-driven approach is the clear choice. If the challenges, requirements and goals are imprecise, undefined and/or changeable, then an agile approach fits better.



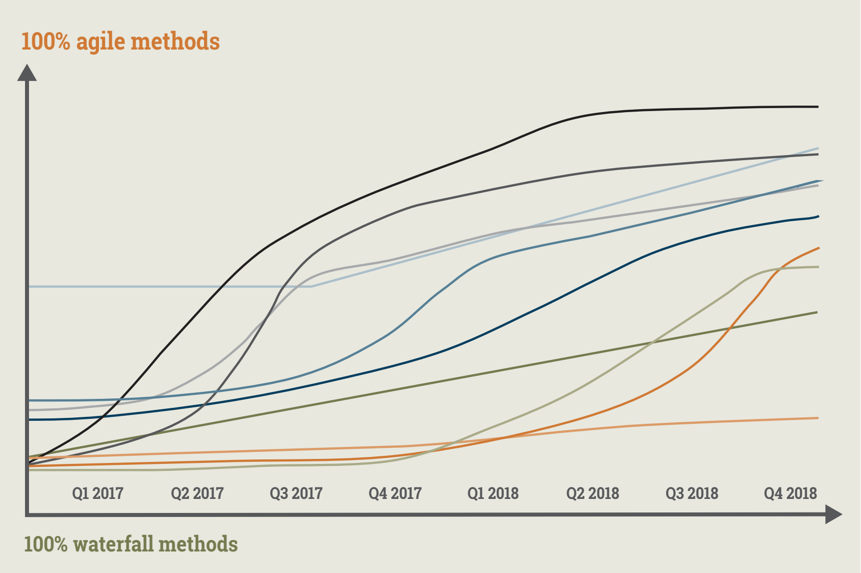
The time set aside for the project also plays a part: It may be very difficult to follow a plan-driven approach when time is short. However, the agile approach can be used for projects with both shorter and longer timelines. See also our infographic, “Agile or waterfall”, which makes recommendations for how to create clarity before jumping into the agile.

IS AGILITY PART OF YOUR STRATEGY?

We have asked the IT managers in a number of larger Danish organisations about their expected journey from traditional to more agile development methods. This has yielded insights into how organisations to a large degree plan to implement agility, while also illustrating fairly major differences in their expectations with regard to speed and goals.

Agility is high on the agenda for many IT organisations as a method to create more efficiency, transparency and – not least – far more embedded IT projects and initiatives in the business. With the agile approach, IT solutions are developed in smaller segments and have a strong emphasis on targeting the requirements, design and functionality of the solution continuously and in cooperation with users. This results in excellent opportunities to achieve better results quicker and to ensure that the solutions meet the immediate needs of the users.

The graphic shows 9 companies' plans for how much agility they wish and aim to implement in their IT projects in 2017 and 2018. Representatives from the businesses drew the graphs in the figure below to illustrate their expectations of their transformation from a traditional approach (lower half) to the agile approach (upper half). At the beginning of 2017, nearly all the businesses had a vast majority of traditional projects, but it is clear that they all aim to move toward more agile methods. One of the organisations has furthermore specified that there will be a difference with regard to whether small projects (such as pilot projects) are being developed, or if large "legacy" systems are involved: The choice of method depends on the nature of the IT system.



How to best make the change?

We have spoken to two different IT organisations that have two different ways of implementing agile development methods. One has a structured, planned administration, while the other has a taken "grassroots-oriented" approach. On the next page, you can see how each of them took on the matter.

COMPANY 1

Start-up and background

We completed a pre-project: An independent survey of the IT organisation's ability to deliver. The survey resulted in a report with a long list of “possible actions”, which pointed in the direction of implementing more agility in approach, processes and tools. The independent survey ensured upper management's (both in the group company and in IT management) acceptance of and support for the process.

Status

We are at work implementing the agile mindset, processes and tools in the IT department itself. Our plans for phase 2 have not been set yet: The implementation of agility should be agile. We do already know, however, that embedding and retrospectives are key elements in the way forward. The next big step is to dissolve the plan-build-run organisation. The plans for the upcoming dissolution has already been revealed to the company to get ahead of rumours and discontent. Many roles will change, including the project manager role. We expect to be up and running with agility in about 3 years. We place a strong emphasis on our transformation leadership: We wish to make it easy for employees to understand our goals, values and process.

Management

We have chosen to implement the first phase exclusively in the IT department with a management group that covers the existing plan-build-run organisation. In phase 2, we bring the business closer with one or more members of the management group.

Staffing

We have previously identified a need to support employees and have employed two agile coaches: One works with the mindset and way of thinking, while the other focuses directly on building and developing a toolbox that supports agile work forms.

COMPANY 2

Start-up and background

We have started using partially agile methods in different contexts without having concrete plans for how far we will go. The embedding takes place continuously. The employees themselves have identified a need to change their methods to yield better, faster progress.

We have inner motivation because the IT function has a reputation for poor quality in its deliveries, and because of the dominant perception of IT as an outdated, arrogant entity with slow processes.

Status

Mixing the two approaches is a challenge. We aim to work with several different development models to meet the characteristics of different needs. We continue to run traditional projects, but many new developments will be conducted as “backlog initiatives”, which have an initiative owner and either a Scrum Master (with tasks conducted by a Scrum Team) or a Delivery Manager (with tasks conducted by selected people in IT). We are now working halfway agile in one specific area – as a kind of pilot and to test how a backlog-driven initiative works. We see it as a kind of “trial and error”.

Management

There is no formal management group. Each individual area manages their own journey towards more agility in close cooperation with management, so the organisation achieves a certain degree of unity.

Staffing

We have an agile coach who works with the agile mindset: Among other things, the emphasis is on only working on relevant problems. But we lack the agile toolbox to support these new work forms.

What is the most important?

There are various commonalities between the approaches of the two companies, such as the underlying premise for the implementation of agile methods being the agile manifesto (see fact box below). If an organisation intends to move towards more agile methods, the first step is to implement the "agile mindset" to create understanding and acceptance of the changes. This demands a strong focus on transformation management: It is about changing a way of thinking and the entire way one relates to IT development, so here we must work on the fundamental assumptions and mindsets of the employees, as well as the organisation as a whole.

THE MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT

We discover better ways to develop software by doing it
and by helping others do it. Through these efforts,
we have learned to value:

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

There is value in the items on the right, but we value the items on the left more.

There is also no doubt that the traditional way of prioritising and managing projects through a centrally embedded PMO function must change. Here we need to produce more agility in order to handle the agility of the IT projects. New methods are needed to measure progress and value, because in an agile approach, IT projects are often executed with firmly defined resources and time frame, but with a variable delivery that can change throughout the lifecycle of the project.

In the agile world, we do not talk about projects, but rather about "value". Value can be created through "initiatives" which can be delivered efficiently and in small doses, so value can be achieved quickly, rather than requiring a large project that demands a lot of time and resources. The organisations – both the IT department and the business – must go from the traditional stance: "We will provide xx at xx date" to the agile: "We will provide "small doses" continuously, and can prioritise at all times". The way of thinking is to be raised to a new level, so that all parties keep their eyes on the highest goal, while accepting that specific details in the solutions will come into place continuously.

Both the organisations are at the beginning of their journey, so time will tell if the top-down (company 1) or the grassroots-oriented approach (company 2) is the most effective. Further down the road, success depends on the organisations' willingness to change and management's ability to secure motivation, show results and create security for the employees. And the way to do this "right" depends to a large extent on the culture, traditions and values of the organisation in question.

You can compare the journey towards more agility with the implementation of Lean principles. In many organisations, it took 10-15 years before principles, mindsets and processes were finally implemented and became a natural part of daily work. It is clear that the implementation of more agility and methods to support agility takes a long time, and that there is no complete recipe for how an organisation can move from traditional development methods to a more agile approach. Goals and methods must be adjusted continuously, and the move must be adapted to the organisation in question.

PARTIAL AGILITY – HOW CAN IT BE DONE?

Transforming an organisation from having a traditional project approach to being 100% agile is a major task, and certainly not all organisations wish to (or are able to) do this. In some organisations only parts of the organisation are interested in being agile, or management may determine that it makes no sense to be agile in all areas. A traditional organisational structure and governance model can result in challenges for agile efforts. On the other hand, if the entire organisation is agile, it can make external cooperation difficult.

There are many good suggestions as to how you can get started reaping the benefits of the agile approach without jumping right into 100% agility. This can be done by picking out some of the agile methods and using them in an approach that includes both agile and plan-driven elements, and that attempt to use the best of both worlds. The mix of traditional and agile methods is called a hybrid model, and can be found in many different forms from organisation to organisation. A good example of a hybrid model is the "Agile Stage-Gate®" model developed by DTU, GEMBA Innovation and DI, in which a number of agile methods and tools are used alongside a more controlled, traditional structure. (Read more at **www.di.dk/asg**).

Working agile and being agile

In other words, there are many ways to use agile – or many degrees of agility. It is a good idea to start with simple elements, so that employees get used to aspects such as the meeting form in a stand-up meeting. At the same time, it is important to continuously assess what is efficient and adds value to each project or team. The use of individual elements can also help create the motivation and interest in the agile methods or tools needed for a transformation process.

The few organisations that aim for a fully agile methodology and mindset face many challenges, in particular because implementation of the agile mindset demands that all the members of the organisation are fully prepared for change – on a mental level. It can be relatively easy for employees to relate to and to grow accustomed to using a range of new tools and methods, but changes to the fundamental mindset and development paradigm requires a high degree of openness and willingness to change.

On a completely concrete, practical level, the geographical distance between an organisation's locations as well as cultural differences between employees may also mean that the implementation of agile methods and process should be adapted carefully. The technical support for telecommuting must also be in place.

We can talk about **working** agile when we use one or more agile methods or tools. If we aim to **be** agile, we need our entire stance on our ways of working to be agile, to change organisational culture and the mindset of the employees. **Being** agile should not be a goal in and of itself, but should be used as a means to gain the even greater advantages of an agile transformation.

There are many organisations and businesses that already mix the two approaches. A study from 2017 illustrates the agile components or methods that are used in the organisations.

90%
DAILY STANDUP

88%
SPRINT/ITERATION PLANNING

85%
RETROSPECTIVE

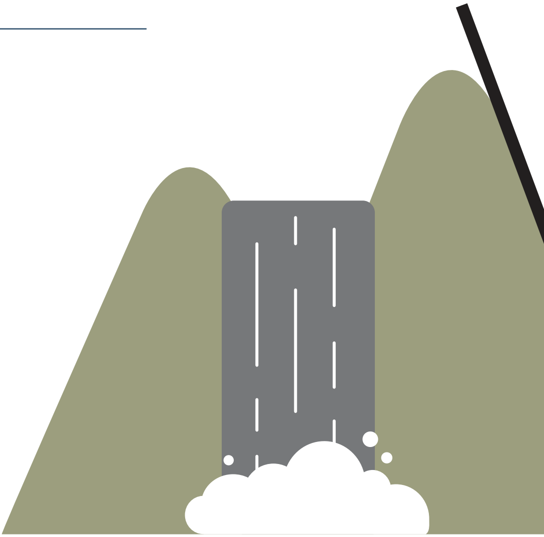
80%
SPRINT/ITERATION REVIEW

69%
SHORT ITERATIONS

Source: www.collab.net

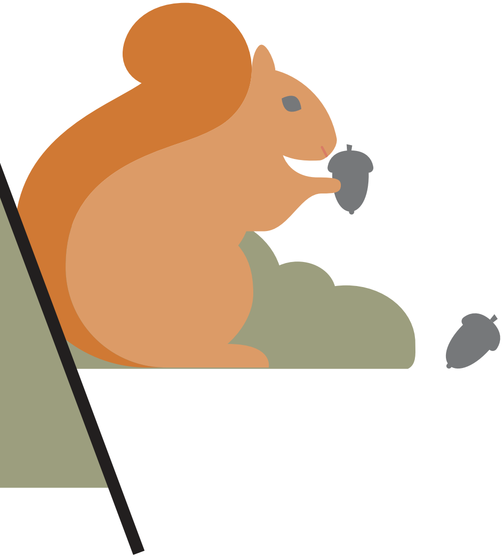
AGILE OR WATERFALL

Create clarity before you make a choice – and make sure to challenge the attitudes you meet



Investigate the business' level of knowledge about agile methods. Maybe you are also facing a change management task if you choose to work agile.

Look at the complexity and size of the project, and whether the goal and scope is already fixed. For example, external factors can cause some parameters to be already predefined.



Find out if the relevant stakeholders are willing and have the time to involve themselves deeply in the development phase.

Make sure to identify what is the highest priority for the business, for example time, cost, quality or scope of the product – and the reasons behind.

Consider if it is important to deliver the full scope at once or if it creates value (and is possible) to continuously deliver functionality.

PROS

AGILE

- The customer approves all phases.
- Frequent, small deliverables that provide value to the customer quickly – and helps to discover errors early in the process.
- (Re)prioritization happens continuously based on the value of the deliverables, prompting scope changes.
- Customer involvement ensures a value-based focus.

WATERFALL

- Early agreement on requirements.
- No need for customer involvement during the development phase.
- The full scope is specified in advance.
- The exact specification of the deliverables decreases the risk of scope creep.

CONS

- Difficult to get full-time allocated resources.
- The customer may not have the time to prioritize the high degree of involvement.
- The employees who are not involved in the project, might find it difficult to understand that the project changes during the process.
- The requirements for fast delivery increase the risk of unfinished deliverables.

- The customer does not see the product until late in the project and there is a risk that it is not satisfactory, and subsequent changes are needed.
- The customer finds it difficult to imagine the final product in the specification phase.
- Late changes increase the risk of exceeding the budget.
- Late changes increase the risk of exceeding the project time plan.

AGILE MISCONCEPTIONS…

Agile is something new…

The Agile Manifest, which is the foundation for agile methods was conceived in 2001. It is not a new area!

Agile equals no documentation…

Rather the opposite: an agile approach requires a focused process of on-going documentation of the frequent deliveries and changes.

Agile methods create value right away…

Maybe… But it requires your organization to be geared for working agile – and the implementation of agile methods and mindset is not an easy task.

Everyone loves agile…

Probably not right away: there is a huge difference in how the different units in an organization position themselves in connection to agile – most often it is far from everyone who loves agile at first.

There is no planning in agile…

There is at least as much planning as in the traditional development methods: planning happens continuously and is repeated, and it depends on the specific situation.

Requirements specification is not needed…

Yes, agile requires you to continuously (re)consider the customer needs. The demands must be handled in an agile and flexible way, so focus always stays on fulfilling the needs of the customer.

SCRUM – FRAMEWORKS AND ROLES

Background and frameworks

Scrum is an agile method for delivery management for several types of projects in various industries. This may refer to technical product development, software development and concept development, etc. The Scrum method contains a specific and well-tested process framework based on dialogue and understanding.

The method draws much of its inspiration from the Lean philosophy, and focuses in particular on getting the best out of the team set to complete a specific task. Scrum is based on teamwork and cooperation, and ensures quick decision-making when the team must respond to changes, new challenges or the inclusion of newly discovered knowledge.

How does a Scrum project start?

Scrum is basically a delivery management methodology. This is why we need the goal to be defined, but the “goal” here does not necessarily have to mean the contents of the final delivery, but rather the solution to the customer’s needs. In Scrum, the delivery is defined and created on the basis of a *Product Vision*. A Product Vision is based on the customer's needs.

Product Vision comes from a “proposer”, who has a need. The proposer may be an internal customer – such as the sales department of a major company requesting that the IT department implements a system that allows them to keep track of entered contracts. The proposer may also be an external customer seeking a product – such as a customer who intends to buy a new type of machine. The proposer is represented by a Product Owner, who is responsible for setting a vision for a product or a service that can improve or secure the foundation of the Product Owner’s business.

SCRUM ROLES

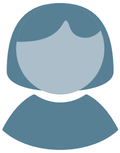
Product Owner: The customer's and users’ voice in the team. The Product Owner defines what is to be developed and in what order. The role is also responsible for getting the most value out of the project as possible, and ensure that it is a profitable investment for the business. The Product Owner creates the vision for the product (*Product Vision*) and owns the *Product Backlog*.

Scrum Master: Facilitates work processes and removes obstacles for the team. The Scrum Master is also responsible for helping the team work in the agile framework, and ensures that the fixed meetings are held (such as Sprint Planning and Sprint Review). The Scrum Master also focusses on the Development Team’s continuous improvement and development.

Development Team: The team that performs the practical “construction work” (programmers, designers, testing resources, etc.) In short: The team consists of all the resources needed to construct a final product. Remember that Scrum is a management method and not the development tool used during the construction process. The Development Team should consist of 3 – 9 people.

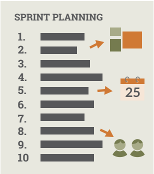
And don't forget...
The Stakeholders: As in all other projects, there is always a group of people who give input or are affected by the project and its results to a greater or lesser degree.

SCRUM – THE PROCESS



1. Product Owner formulates the *Product Vision*, making it clear through a number of feature descriptions. The descriptions are based on the use cases the upcoming product will create, known as *User Stories*.

2. The Product Owner establishes a prioritised list with the desired User Stories in relation to what gives the most value to develop first. This list is called the *Product Backlog*.



3. The prioritised User Stories from the Product Backlog are converted to a ‘To-do list’ for the team that is to develop functionality. This list is a Sprint Backlog and contains the tasks that the Scrum Team is obligated to complete within a given time frame, known as a *Sprint*. Estimation of the task scope and planning of the execution of the work is done in cooperation between the Product Owner, *Scrum Master* and the Development Team.
The planning itself is called *Sprint Planning*.

4. The Development Team, led by the Scrum Master now works determinedly in a Sprint. The length of the Sprint is determined on project start and typically lasts from one to four weeks. The team works on developing the desired functionality into a fully-functional product, which in principle is ready to be delivered for use (*Minimum Viable Product (MVP)).*

5. Every day, the Scrum Master holds a stand-up meeting: *Daily Scrum meeting*. Here, the team focuses on three things – both on the team and individual level:

   - What was made yesterday?
   - What will be made today?
   - What could prevent the team from making progress today?

6. At the end of each Sprint, a Sprint Review meeting is held, which contains a review/demonstration of the developed feature for the people (product stakeholders) who must confirm the developed feature. In this meeting, the focus is on the product itself.

7. At the end of each Sprint, a *Sprint Retrospective* meeting can be held. The key question is: How did the Sprint go, and what can be improved next time? In this meeting, the focus is on the process.

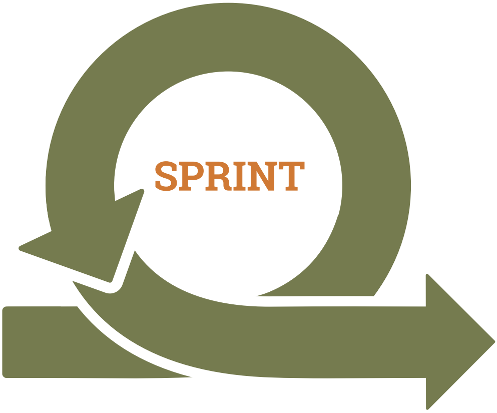
8. The cycle with points 3-7 is an iterative process that repeats, where topics (User Stories) are prioritised and selected from the Product Backlog again and again. This continues until the total project delivery has been completed to a level at which the Product Backlog no longer adds value to the Product Owner. In principle, a Product Backlog does not close as long as the product is live.

The above is a greatly simplified overview using a product with very little complexity. If complexity increases, the Scrum model will be open to expansion with a roadmap for the product, which instructs the Scrum Team and other stakeholders on the vision for the product's lifespan, etc.

Similarly, there may be a need to expand the Scrum model with a plan for when the achieved deliverables should be "released" (and put into operation). A release plan may also help facilitate the prioritisation of the items in the Product Backlog: You develop the things that give the most value first.

There is a lot of project-oriented hands-on expertise behind several of the elements. For example, the work with the Product Backlog demands that one is able to optimise and refine it, while at the same time ensuring that the descriptions (Use cases) are understandable and make sense to the Development Team. The role of Product Owner is a discipline of its own. Today it is possible to become internationally certified as a Product Owner.

If you use Scrum for exceptionally large projects, it may be a good idea to take a closer look at the agile framework tool called SAFe. We describe the basic principles in the final article in this booklet.



DAILY SCRUM
– IN SHORT

What is it?

- A daily 15-minute meeting
- The meeting is held every day throughout the entire Sprint
- The meeting is held at the same time and place every day
- Each individual participant answers three specific questions
- The participants are: the entire Development Team, Scrum Master and if applicable, the Product Owner

What does it give?

- All members of the team always know how far work has come
- Progress is made visible and motivating
- Plans for the Sprint can be adapted continuously based on the meetings
- The team takes responsibility and becomes autonomous
- Builds trust between team participants

Remember:

- Daily Scrum is not a status meeting
- The meeting is not held for the Scrum Master’s sake
- This is not a place for technical discussions
- Whenever possible, the meeting is held in the team's workroom
- All three questions must be answered by all participants

THE THREE QUESTIONS:

- What did I work on yesterday?

- What am I working on today?

- What obstacles (if any) could, keep me from completing my task?

3 TECHNIQUES **FOR ESTIMATION**

Estimation in an agile team or project group can be difficult. Often, discrepancies may be due to differing perceptions of what the task involves. Therefore, the foundation for a solid estimation process is a clear and easily understood description of what the element or task is about. It is important that all relevant team members are heard and can make their suggestions, and there is a variety of ways to make sure this takes place.

PLANNING POKER

- Each individual team member has a set of 9 cards marked 1, 2, 3, 5, 8, 13, 21 (the first numbers in the Fibonacci sequence), “divide [scissors]”, “?” and “∞” (the infinity symbol)

- The Product Owner describes the functionality of the element that is being estimated

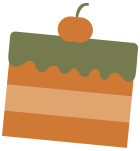
- Each participant selects the card that represents the work effort that must be invested to develop the functionality

- All participants show their cards at the same time

- The selected estimates are discussed as the people with the highest and lowest estimates make arguments in a general discussion (also known as the Delphi method). The process is repeated until a pre-determined level of consensus is met (for example that there is only one estimate between them and the majority decides the outcome afterwards)

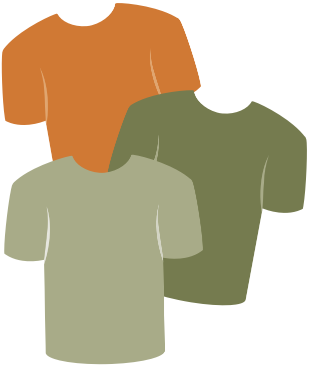
- Along the way, clarifying questions appear, requiring the Product Owner to explain the element further, which helps the team in the estimation process

CAKES



- Cards with various cakes can be used instead of the Fibonacci sequence, e.g. cookies, muffins, sheet cake, layer cake, cake buffet. The cards “divide [a knife]”, “?” and if applicable, “∞” (infinity symbol) should also be part of the set
- Each cake represents a number of “story points”, which tell us something about how much work is needed to develop the individual “story” (element) in the list
- The estimation process takes place as described above
- When the element is developed, the team celebrates by eating the cake that represented the workload

T-SHIRTS



- A simpler version with a rougher estimation uses T-shirt sizes. Here, each individual card has a size description: S, M, L, XL. The cards “divide [scissors]”, “?”, and if applicable, “∞” (infinity symbol) should also be part of the set
- The estimation process takes place as described under “Poker Planning”

INTRODUCTION TO **SCALED AGILE FRAMEWORK (SAFe)**

What is SAFe?

Scrum has many advantages and can be an efficient development methodology, but it does not describe how to scale to larger projects and ensure that several agile teams work across an organisation. Scaled Agile Framework (SAFe) is a framework for both product and IT development created by **Scaled Agile, Inc**. SAFe defines a range of structures and processes that serve as a scalable approach to Lean-agile processes. SAFe enables organisations to use agile methods (Scrum) on an organisational level, so that cooperation and deliverables can be matched and coordinated through many different agile teams.

Scalable organisation

SAFe defines four levels for the agile organisation as well as a fundamental level. SAFe supports everything from the development of smaller, disparate solutions to very large, complex systems across multiple business sections:

- **Team level:** SAFe is based on agile Development Teams, which typically follow the Scrum method (or Kanban). Each team is responsible for defining, developing and testing “user stories” (see the glossary) from their backlog. The team delivers functionality every time a Sprint is complete.

- **Program level:** SAFe teams are gathered in virtual programs called Agile Release Trains (ART). An ART is a long-lasting organisation consisting of several SAFe teams and relevant stakeholders, who together plan, execute, test and adapt deliverables surrounding a value stream in the business.

- **Value stream:** The development of large, complex systems can be split into value streams, which gather several ARTs for a given solution. Deliverables are synchronised throughout a group of ARTs.

- **Portfolio level:** Here, a range of value streams are organised and financed, supporting the business's strategic targets, defined in strategic terms. Portfolio management finances solution development through lean-agile budgeting, ensuring governance and coordination across value streams, where this is needed.

- **The foundation:** Here we gather some fundamental processes, elements and people that ensure, maintain and expand the agile way of thinking in the organisation. This may be Lean-Agile managers, Best Practice Communities, SAFe values, SAFe principles, etc.

The above is based on the whitepaper produced by Scaled Agile INC. **(www.scaledagileframework.com)**.

SCRUM GLOSSARY

Artefacts

Artefacts are intended to visualise status and progress. We work with three fundamental artefacts in Scrum: Backlog, Backlog Refinement and the Burndown chart.

Backlog

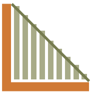
See Product Backlog or Sprint Backlog, depending on the context.

Backlog Refinement

Working to further develop and refine the Product Backlog. Work may encompass changes to sequence, adding or removing items, re-writing, estimating and re-estimating or dividing existing items. Backlog Refinement meetings ensure that the Product Backlog is kept up to date and relevant with regard to the customer’s needs. It must contain enough tasks, described in the proper format, for the team's next Sprints.

Burndown chart

A visual presentation of how much work needs to be done (vertical axis) compared with time consumed in the project (horizontal axis). This can be both the Sprint Burndown chart and the Product Burndown chart.



Daily Scrum meeting

The daily stand-up meeting, which is to be kept short (15 minutes). Those present literally stand up for the meeting, and others participate from a distance through means such as Skype. Each individual member of the Development Team explains:

1. What did I do yesterday that helped the Development Team meet the Sprint Goal?
2. What will I do today to help the Development Team meet the Sprint Goal?
3. Do I see any impediment that prevents me or the Development Team from meeting the Sprint Goal? (It will be the Scrum Master’s task to remove

these obstacles). The meeting can easily result in the Sprint Backlog being updated. The meeting is also called a “daily stand-up”, “daily meeting”, “huddle” and “roll-call”.

DoD

Definition of Done. A list of criteria that must be met by each individual element before it can be considered done and ready for implementation. DoD is thus a list of quality requirements for everything being developed. The Development Team owns the list and ensures transparency and a shared understanding of how a finished element looks – it is important that all stakeholders: Product Owner, Scrum Master, Development Team and customers agree on when something is finished.

DoR

Definition of Ready: The set of conditions that must be met before an element is ready to be included in a Sprint related to Sprint planning. An example could be that the element must be thoroughly described through a detailed User Story.

Epic

An Epic is a major User Story, too wide-ranging to be developed in a single Sprint. Epics must be broken down into smaller User Stories before they can be placed into a Sprint.

Estimation

A quantitative description of how much work is needed to complete a concrete element from the backlog. There are many ways to estimate, such as standard working hours or Story Points. Estimation takes place in the team to ensure responsibility is taken, and it can be done through exercises such as Planning Poker. Read more in our article about “3 Techniques for estimation”.

ETC

Estimate to Complete: An estimate of how many hours are needed to complete a concrete task.

Fail-fast (fail-small)

An approach to development where small experiments are performed to get quick feedback, which can then be assessed and taken into account at once. If you stand before major uncertainties, it may often be better to try an idea with minimal risk and costs, so that these can be rejected before spending too much time and money on them. The approach is also known as fast-feedback or learn-fast.

INVEST

An acronym used to assess the quality of a User Story:

- Independent
- Negotiable
- Valuable
- Estimatable
- Small or appropriately sized
- Testable

LeSS

Large Scale Scrum is a method for scaling Scrum, so the solution can be used in connection with large-scale development. The method, which was created in 2013 by Craig Larman and Bas Vodde, consists of principles, rules, guidelines and experiments. LeSS focuses on understanding a single Scrum Team first, then scaling up. The scaling can happen so that the collective Development Team consists of 8 Scum Teams (LeSS) and up to thousands (LeSS Huge) – all working on a single product. Take a look at the LeSS home page: **www.less.works,** if you would like to know more.

MVP

Minimum Viable Product: A product that has the exact (no more and no less) functionality that allows the team to implement it. MVP is typically the first thing to come out of the first Sprint and is the goal of the Sprint itself.

Planning Poker

A tool that can be used for estimation. The Product Owner presents an element (a piece of functionality) with a view towards the team estimating how much work is needed to develop the described item. Each individual team member has a number of Planning Poker cards, each with a number of hours or points written on them. Each individual team member makes a suggestion for an estimate by choosing the card they feel gives the best estimate. All members show their suggestion at the same time. Afterwards, the team members with the highest and lowest suggestions, respectively, explain their reason for choosing these estimates. This is followed by a team discussion. The process using suggestions from all team members continues until the entire team agrees on an estimate that everyone can support. Estimates can be made in many ways, but the key point is that everyone can accept the estimates and has properly familiarised themselves with the reasons behind them.

Prioritisation

The Product Owner prioritises, from Sprint to Sprint, the Product Backlog, to decide the most important thing to make. This prioritisation will decide the key content of coming Sprints.

Product Backlog

A list of all functionality the product as a whole is to include. The Product Owner owns the Product Backlog. The elements in the list may take the form of User Stories, use descriptions, or be described in other formats.

Product Burndown chart

A graphic overview that shows how much remains in the production of the total product/service. It focuses on how much remains until the product is ready for full delivery.

Product Owner

Is a part of the Scrum Team and is the voice of the customer and the user. He/she defines what is to be done and in what order. The role is also responsible for getting the most value out of the product and ensuring that it is a profitable investment. The Product Owner creates the Product Vision and owns the Product Backlog.

Product Vision

A description of the future condition achieved by developing and implementing the product. A good product vision is simple, easy to understand and creates shared focus for those working on the development of the product.

SAFe

Scaled Agile Framework is a scaled Lean-Agile method developed by Dean Leffingwell in 2011. Read more on the official page for SAFe: **www.scaledagileframeworks.com.**

Scrum Master

The Scrum Master facilitates Scrum work processes and is responsible for removing obstacles for the team when it has to live up to delivery requirements in each Sprint. The Scrum Master is also responsible for helping the team work in the agile framework, ensuring that fixed meetings are held (such as Sprint Planning and Sprint Review), and must also make sure that the relationships in the team are functional.

Scrum Team

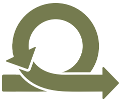
The Scrum Team consists of a Scrum Master, a Product Owner and a Development Team. Together, they are responsible for delivering the functionality promised in the individual Sprints, that the quality is as agreed and that the delivery is ready on time.

Scrum workspace

Supporting the agile work processes requires an overview that ensures that everyone can be oriented and direct their attention to the same goal. This overview is often made by using a large blackboard or wall where each task in the individual Sprint is shown. It must be clear who is working on the task and what phase the task is in, such as design, development, unit test, system test, user test or documentation. A simpler version can also be selected, where tasks are placed in columns labelled “not started”, “in progress” and “completed”. The blackboard can also include a Sprint Burndown chart to show total progress – and leave space for unexpected needs, problems or surprises that may appear...

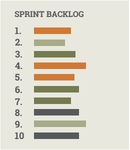
Sprint

A time period, typically 2-4 weeks, that defines how much work can be put into the product.



Sprint Backlog

A list containing the functionality that has been agreed to be produced in the Sprint. The list is based on the team's estimates for how much time is to be spent on completing each individual element. The Development Team owns the Sprint Backlog, which is updated throughout the entire Sprint, so it always gives a clear and visible representation of the remaining work. The team can choose to break the elements in the backlog down into smaller tasks, which are completed in the Sprint.



Sprint budget

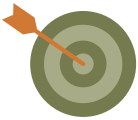
The number of Story Points (or similar) that can be produced in a Sprint.

Sprint Burndown chart

A visual overview showing how much remains of the tasks in the current Sprint. Scrum does not focus on what has been achieved, but rather on what remains when reporting the Sprint.

Sprint Planning

The Sprint Planning meeting takes place at the beginning of each individual Sprint. The purpose of the meeting is to set the goals and tasks for the Sprint.



Sprint Retrospective

The Sprint Retrospective is a meeting. Here, the team takes a collective look back at the Sprint, and also uses the information from the Sprint Review. The purpose of the Sprint Retrospective is to improve the efforts and performance of the team while at the same time implementing better ways to complete their tasks. This may refer to work processes, communication, mutual relationships, methods, tools, and more.

Sprint Review

The Sprint Review meeting takes place after each Sprint. The Development Team shows what they have completed in the Sprint through means such as a demo of the produced deliverables, and the products are evaluated in cooperation with the other stakeholders (the Product Owner and the customer).

Sprint Task

A further breakdown of User Stories from the Sprint Backlog to a description of the actual tasks that are to be started by the Development Team. The Sprint Task describes **how** to create what the User Story covers.

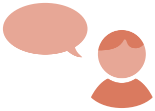
Story Point

The work required to develop and implement each individual User Story can be estimated by using the Story Point method. Each User Story is estimated to a number of points depending on how complex the functionality is to develop. From here, it is possible to assess how demanding the various User Stories are in relation to one another.

Stretch tasks

Extra tasks outside of a Sprint, but that the team can take on if they are able to complete the tasks included in the Sprint.

User Stories



User Stories describe what the functionality of the upcoming product will mean to the user's world. The reader gains an understanding of how the functionality creates value for the user.

Velocity

The productive ability per Sprint, measured over time, represented in Story Points, for example.

Discover and learn much more at

www.kaastrupandersen.dk

kaastrup | andersen

Vestre Ringvej 45A
DK-8600 Silkeborg

Park Allé 382
DK-2625 Vallensbæk

Phone: +45 70 27 77 19
info@kaastrupandersen.dk