

# Agile methodologies

*This is a compilation of content derived from several public sources which are indicated. The only purpose of this document is to be a reader in preparation of the EXIN Agile Scrum Product Owner certification.*

Since the publication of the Agile Manifesto back in 2001, many organizations have adapted different Agile methodologies to enable their teams to achieve their goals and deliverables. The most famous methodology is Scrum, which is widely used in all kinds of companies. In the early days, these Agile practices were only applied by IT departments. By contrast, in 2019, Agile methods have become broadly used in a wide range of different non-IT business areas.

This article will provide an overview and brief explanation of the most practiced Agile methodologies aside from Scrum:

- Crystal Methodologies
- Extreme Programming (XP)
- DSDM
- LeSS
- SAFe
- Kanban

It is important to keep in mind that although these methods are all different, they have 3 things in common. Open communication, togetherness and flexibility. These elements are shared by all of these methods because they are at the core of the Agile Manifesto.

Copyright © EXIN Holding B.V. 2019. All rights reserved.

EXIN® is a registered trademark.

Crystal Methodologies have been developed by Alistair Cockburn

DSDM® is a Registered Trade Mark of the Agile Business Consortium Limited in the United Kingdom and other countries.

LeSS has been developed by Bas Vodde and Craig Larman

© Scaled Agile, Inc.

Kanban has been developed by Taiichi Ohno

No part of this publication may be reproduced, stored, utilized or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written permission from EXIN.

## Crystal Methodologies

The Crystal family is a group of **lightweight** Agile methodologies. They are considered lightweight because these methodologies consider process of secondary importance and instead place the emphasis other elements.

These elements are:

- People;
- Interaction;
- Community;
- Skills;
- Talents;
- Communications.

Crystal methodologies were developed by Alistair Cockburn in the 1990's. They were created as a result of research conducted by Cockburn that showed that development teams weren't using formal methodologies as they were intended to be used, but still they were delivering successful projects. Cockburn differentiates between some important aspects:

- Methodology: a set of elements (e.g. practices, tools);
- Techniques: the skill areas (e.g. developing use cases);
- Policies: definitions of how the organization must behave.

As a result of his research, Cockburn defined the behavior of people in teams as follows:

- "People are communicating beings, doing best face-to-face, in person, with real-time question and answer."
- "People have trouble acting consistently over time."
- "People are highly variable, varying from day to day and place to place."
- "People generally want to be good citizens, are good at looking around, taking initiative, and doing 'whatever is needed' to get the project to work."

Crystal methodologies are divided into 8 different colors. The colors are used to indicate the 'weight' of the methodology. The list starts with Crystal Clear that is suitable for projects with up to 6 people and carries on through to Crystal Diamond or Crystal Sapphire that is suitable for mission-critical projects that involve a potential risk to human life.

An important point that requires attention as well, is that there are 7 common properties shared by all of the Crystal family. These properties are:

### **Frequent delivery**

Regular releasing of iterations of the software/product that is in development.

### **Reflective improvement**

The project team is invited to think about how to improve their processes from time-to-time. This not only provides a break from development tasks, but also helps to improve their efficiency.

### **Close or osmotic communication**

Communication must flow through the project team. To ensure this happens, the project team should be together in the same room so that communication that supports the progress of the project is encouraged.

### **Personal safety**

Everyone in the team must feel they are in an open and safe environment where they are free to speak up. Negative responses, such as being ridiculed when they ask a question or suggest an idea must be

completely discouraged. This is because people must be able to trust each other and anyone who is the target of negativity is likely to not participate proactively anymore.

**Focus**

Keeping focus is crucial for success. When we talk about focus in Crystal methodologies, it refers to two things. In short, they are progress and direction. By progress, we refer to dedicating enough time to a task in a project that progress will be made. Whilst direction refers to the direction the project is heading in.

**Easy access to expert users**

An actual/real-life user of the new feature that is being developed must be reachable by the project team to answer any questions, help with solutions to problems and so on. This expert can help the project with their hands-on experience and, ideally, they should be available often (including during meetings or through phone calls).

**Technical environment with automated tests, configuration management and frequent integration**

Frequent integration and testing is essential so that any issues (errors, bugs etc.) can be spotted early. If integration is continuous it prevents problems from growing as they are resolved early on.

Source: [Wikiversity](#)

## Extreme Programming (XP)

XP is a software development methodology that puts the focus on customer satisfaction. This guarantees that what is being produced is based on the needs of the customer. One of the ways it does this is by transforming user stories into valuable business assets.

One of the defining characteristics of the XP methodology is continual and open communication. Without this important aspect, the practice would not work in line with expectations. An important benefit of this style of communication is an increase in confidence for both the customer and the project team.

XP was created to answer a need in the market to accommodate the continuous changes that service providers were facing while developing a new product or service. Its success can be attributed to the main communication and collaboration aspect between all parties that are involved in the project. This means that customers, managers, project teams and anyone else involved will work together from the moment the scope is set, the stories defined, right up to testing and the final release.

The methodology was designed for small project teams, although large teams are using XP and have reported success as well.

How does Extreme Programming (XP) work?

## Extreme Programming (XP) at a Glance

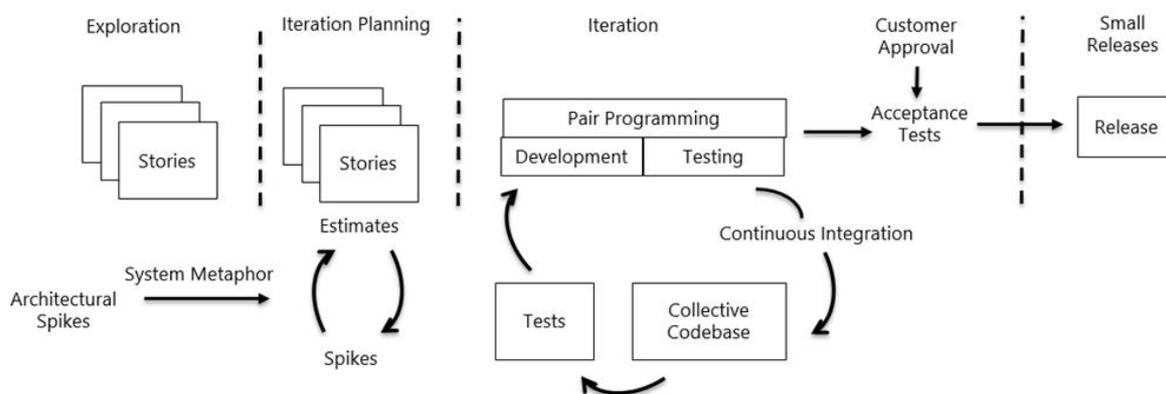


Fig 1 – [Extreme Programming at a Glance](#). Source: (J.D Meier)

The first step in XP is the collecting of user stories as well as conducting spike solutions for those that could represent a risk.

After the initial steps, a release planning meeting must be scheduled. At this stage, it is important to invite all of the required people. This can include customers, project team, and managers. The main goal at this stage is to define a common goal that is acceptable to everybody. This meeting and setting of the goal is followed by the iterative planning meeting in order to agree on the next releases.

With just these few steps, the project team is already moving in the direction of developing the required features.

## The Rules of Extreme Programming

The rules of Extreme Programming are documented on [ExtremeProgramming.org](https://extremeprogramming.org). The rules comprise:

### Planning

- User stories are written.
- Release planning creates the release schedule.
- Make frequent small releases.
- The project is divided into iterations.
- Iteration planning starts each iteration.

### Managing

- Give the team a dedicated open work space.
- Set a sustainable pace.
- A stand up meeting starts each day.
- The Project Velocity is measured.
- Move people around.
- Fix XP when it breaks.

### Designing

- Simplicity.
- Choose a system metaphor.
- Use CRC cards for design sessions.
- Create spike solutions to reduce risk.
- No functionality is added early.
- Refactor whenever and wherever possible.

### Coding

- The customer is always available.
- Code must be written to agreed standards.
- Code the unit test first.
- All production code is pair programmed.
- Only one pair integrates code at a time.
- Integrate often.
- Set up a dedicated integration computer.
- Use collective ownership.

### Testing

- All code must have unit tests.
- All code must pass all unit tests before it can be released.
- When a bug is found tests are created.
- Acceptance tests are run often and the score is published.

Source: [extremeprogramming.org](https://extremeprogramming.org)

## DSDM

Dynamic Systems Development Method or DSDM is a framework that focuses on the full project lifecycle. This approach dictates that only the minimum work will be done in every step, in order to move forward for the next item. The mindset behind this is that continuous change is a natural part of projects. It is a well-known fact that business requirements can change at a moment's notice, so by doing only the necessary work for the required step to be considered completed saves the project team's effort, resources and time.

The DSDM framework can be used for different applications in organizations. From the development of a new product (or service) to financial departments. Any part of the business can benefit from it.

### Principles of DSDM

1. Active user involvement is imperative
2. DSDM teams must be empowered to make decisions
3. The focus is on frequent delivery of products
4. Fitness for business purpose is the essential criterion for acceptance of deliverables
5. Iterative and incremental development is necessary to converge on an accurate business solution
6. All changes during development are reversible
7. Requirements are baselined at a high level
8. Testing is integrated throughout the lifecycle
9. A collaborative and co-operative approach between all stakeholders is essential

Source: [Agile Business Consortium](#)

### The DSDM Framework Phases

The DSDM Framework comprises 6 phases. This consists of a pre-project and post-project phase and the project phase which has four main phases: Feasibility, Foundations, Evolutionary Development and Deployment.

#### Phase 1

##### **Pre-Project**

As part of preparation for the project, a clear objective is defined. The pre-project phase helps to ensure that projects are set up correctly and that only the right projects are started.

#### Phases 2-5

##### **Feasibility**

As suggested by the name of this phase, the focus of this part is whether the project in question is feasible (from both a technical and cost-effectiveness perspective). This phase should not take up too much time – only enough to be able to decide whether looking into the project is worthwhile or that it is unlikely to be viable.

##### **Foundations**

During this phase, the goal is to understand the scope of the work. This includes the how, who, when and where of carrying out the actual work. The project lifecycle is also determined through looking at how the DSDM process will be applied.

### **Evolutionary Development**

The Solution Development Team will apply practices including Iterative Development, MoSCoW, or timeboxing to evolve the solution. The aim being to further evolve the solution so that it is accurate, is built in 'the right way' from a technical perspective, and meets business needs. The team will iteratively develop and test continuously as the project progresses.

### **Deployment**

This phase focuses on 3 main activities: Assemble, Review and Deploy. The aim of this phase is to release a final solution (or a subset of it). The project is formally closed after the last release.

## Phase 6

### **Post-Project**

During the post-project phase, checks are conducted to see how well business benefit expectations have been met in the form of a Benefits Assessment.

Source: [Agile Business Consortium](#)

## LeSS

LeSS stands for Large-Scaled Scrum. This is a method that can be applied when multiple teams are working together on one product or service that is being developed.

LeSS can be applied in cases when:

- Multiple teams are required for a specific development;
- Multiple teams work together towards one common goal;
- Multiple teams work at the same time on a product or service.

If your project does not fit these requirements, then it is better to use standard Scrum.

### The 10 Principles of LeSS

#### **Large-Scale Scrum is Scrum**

LeSS is still Scrum, however it is scaled for large products or services development. LeSS provides a set of rules combined with guides to apply Scrum in a multi-team context. It is not a new Scrum framework.

#### **Transparency**

The main outcomes must be visible to all parties. Creating short feedback loops increases transparency. Transparency is essential to ensure adaptive control and improvement is possible.

#### **More with LeSS**

This principle is central to LeSS and acknowledges that complex organizational solutions are disadvantageous and can cause more problems than they solve. It aims to remove complexity by adopting simpler and different solutions to problems.

#### **Whole product focus**

In LeSS the focus is on the complete product. This is because individual contributing parts of the product have no actual value until they are put together to create the final product. After all, customers buy a whole product, not its parts. Retaining focus on the 'whole' is one of the biggest challenges in large development groups.

#### **Customer centric**

As with product focus, the large scale of LeSS poses a risk to customer satisfaction. In LeSS the Product Owner functions as a connector of customers/users and project teams. To ensure the customer remains the focus, teams are organized based on end-to-end features and components.

#### **Continuous Improvement towards perfection**

As one of the two pillars of Lean Thinking, the continuous improvement towards perfection is also a part of LeSS. In LeSS change is expected, continuous and embraced.

#### **Lean thinking**

Respect for People and Continuous Improvement are central concepts from Lean adopted by LeSS.

#### **Systems thinking**

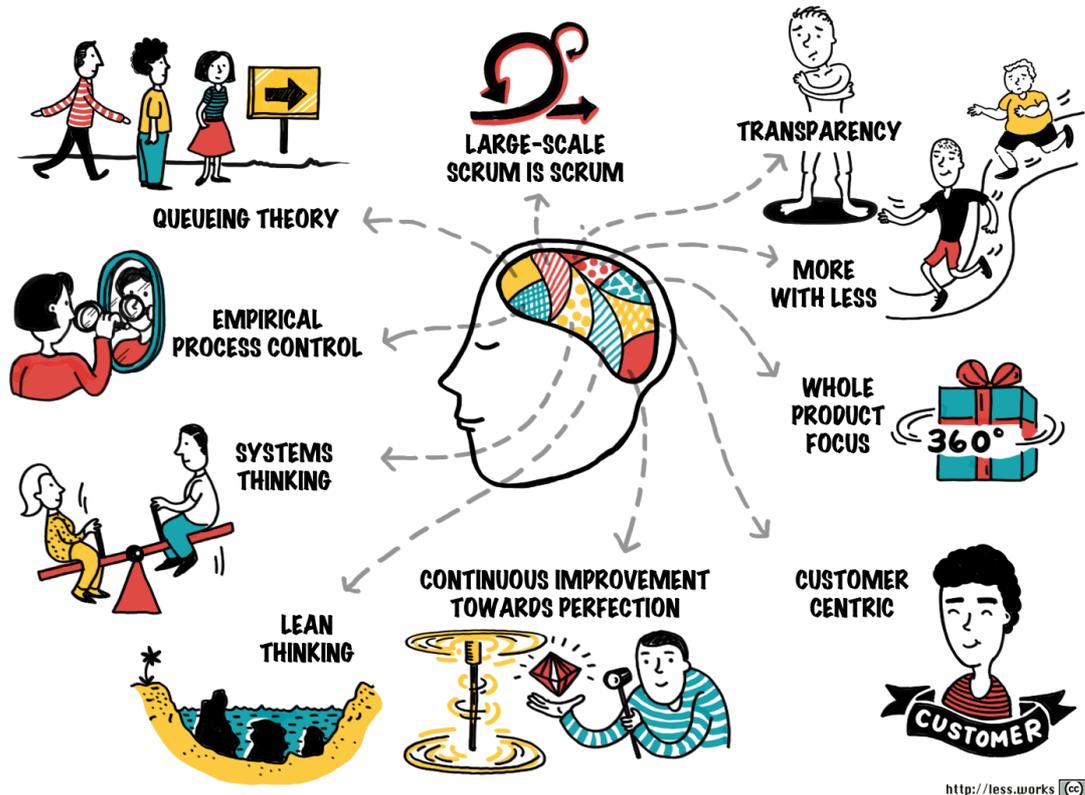
This is the principle that long-lasting systemic improvements are preferable over quick-fix-solutions.

#### **Empirical process control**

This Scrum concept is applied in LeSS to ensure that teams have "just enough process". This way they do their work in a cycle of transparency, inspection and adaptation.

#### **Queueing theory**

This theory of how items move through a system queue is a thinking tool that can be used to improve the ability to handle big batches of work. This is especially relevant to Large-Scale Scrum.



<http://less.works> (CC) BY-ND

Fig 2 – Principles of LeSS. Source: The LeSS Company B.V.

Source: [LeSS Works](http://less.works)

## Scaled Agile Framework® (SAFe®)

The Scaled Agile Framework is a freely available body of knowledge that was created to address issues that are encountered when scaling Agile beyond a team. This enterprise-scale development methodology combines principles from Lean and Agile. The organization and workflow patterns in the framework have been designed to support organizations in scaling their Lean and Agile practices.

### The 9 SAFe Lean-Agile Principles

1. Take an economic view
2. Apply systems thinking
3. Assume variability; preserve options
4. Build incrementally with fast, integrated learning cycles
5. Base milestones on objective evaluation of working systems
6. Visualize and limit work-in-progress, reduce batch sizes, and manage queue lengths
7. Apply cadence (timing), synchronize with cross-domain planning
8. Unlock the intrinsic motivation of knowledge workers
9. Decentralize decision-making

### The SAFe framework

The [Scaled Agile Framework website](#), offers this overview of SAFe 4.5:

In SAFe version 4.5, there are four configurations: essential, portfolio, large solution and full:

- **Essential SAFe** is the most basic configuration. It describes the most critical elements needed and is intended to provide the majority of the framework's benefits. It includes the team and program level (which it calls Agile release trains or ARTs).
- **Portfolio SAFe** includes concerns for strategic direction, investment funding, and Lean governance.
- **Large Solution SAFe** allows for coordination and synchronization across multiple programs, but without the portfolio considerations. In earlier versions of SAFe, this level was referred to as value stream.
- **Full SAFe** combines the other three levels.

Source: [scaledagileframework.com](https://scaledagileframework.com)

## Kanban

Kanban is a technique that has been popular in manufacturing for a long time and has recently been used in IT projects. However, what people mean by “Kanban” in IT environments is usually a development method that is a combination of some Scrum elements with the Kanban technique. This can be called ScrumBan, even though the word ScrumBan is also used to refer to multiple things. Regardless of the labels, let’s start by taking a look at the Kanban technique, and then we’ll look at the development approach that is based on that.

There are different ways to describe Kanban. One of them is with the following rules:

1. Work should be visualized
2. Work in progress (WIP) should be limited
3. Work should be pulled instead of pushed

### Visualizing

Visualizing is really helpful because:

- It creates transparency and, therefore, improved feedback and collaboration
- It creates more control

So, we prepare a Kanban board and visualize the work steps and the work items. The minimum steps we need are to do, doing, and done. You can have other steps too; for example, some people reflect their Definition of Done on the board:

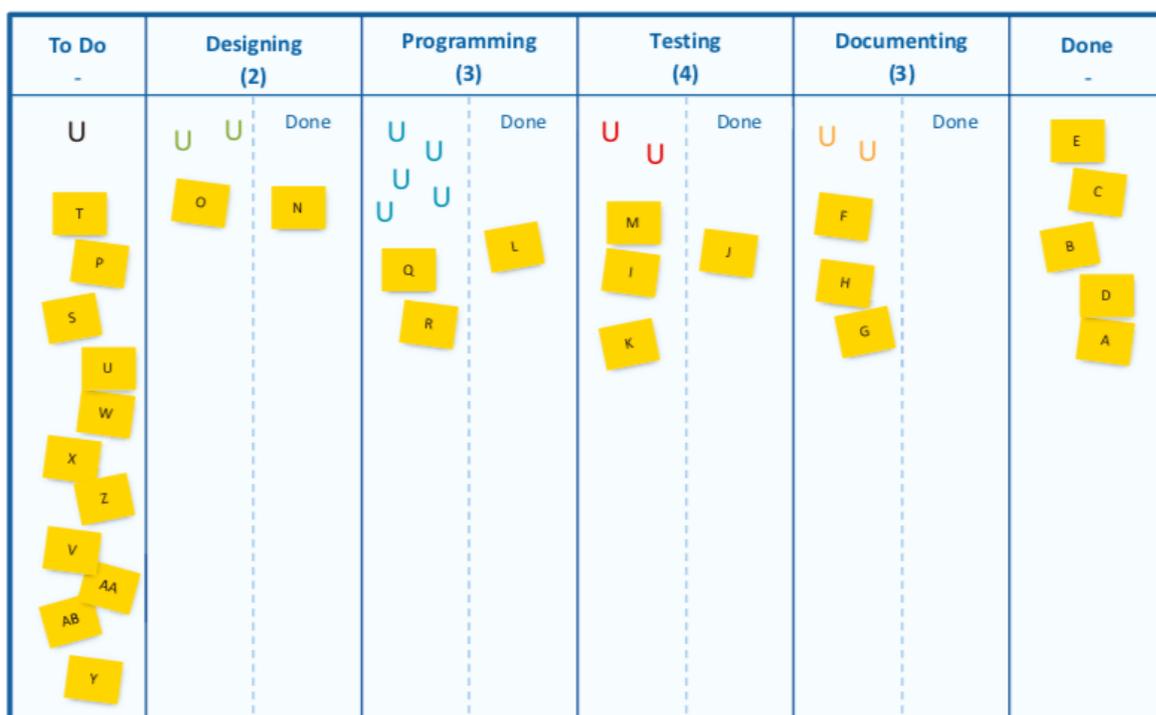


Fig 3 – [Kanban Board](#). Source: [Agile Scrum Handbook \(EXIN\)](#)

In this example, there are 6 main columns:

- To Do
- Designing
- Programming

- Testing
- Documenting
- Done

Source: [Agile Scrum Handbook](#)

## Sources

Crystal Methods - [Wikiversity](#)

Extreme Programming – [extremeprogramming.org](#)

DSDM – [Agile Business Consortium](#)

LeSS – [LeSS Works](#)

SAFe - [scaledagileframework.com](#)

Kanban – The section on Kanban is taken from the [Agile Scrum Handbook](#) with permission from van Haren Publishing.

## Contact EXIN

[www.exin.com](http://www.exin.com)

