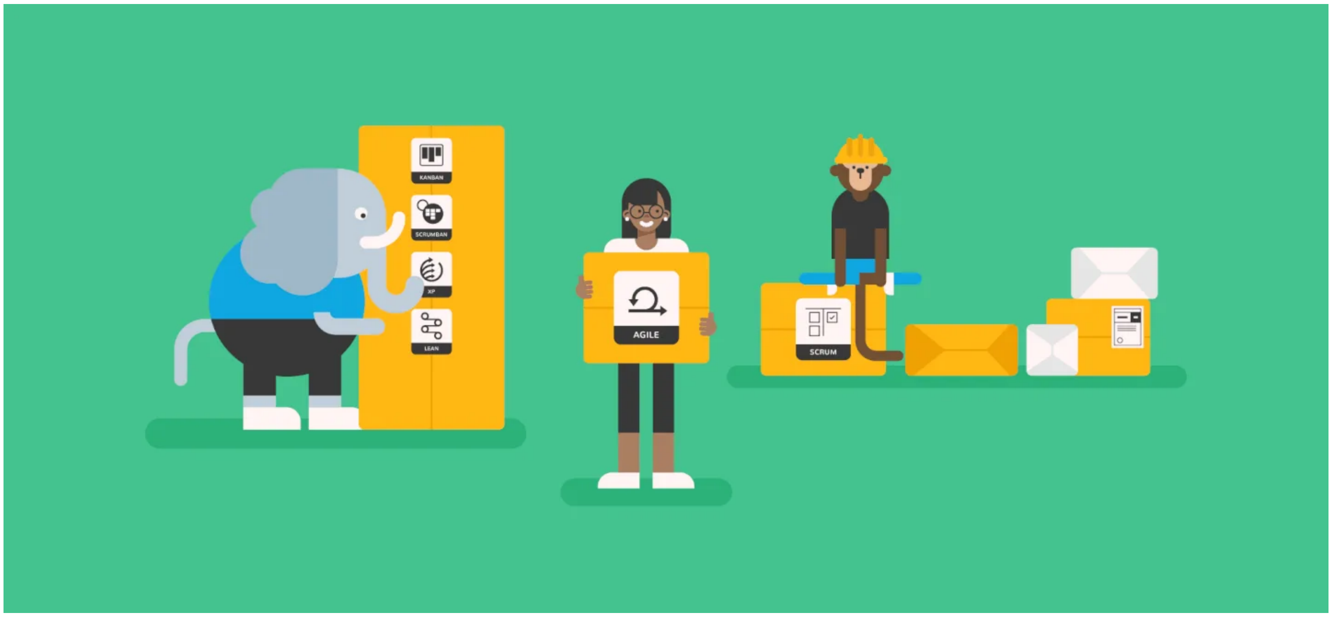
41 Comprehensive Agile Methodologies For Project Managers To Know

QUICK SUMMARY

There are many, many methodologies that fall under the agile umbrella. Here are just 41 of them: get the details on how the top agile methodologies work, and where to find more info.



Some years ago, you could say "*Scrum is agile*" and ask "*is agile Scrum?*". Now we know there is more flesh on the bones.

At this moment there are more than fifty known and less known agile approaches, frameworks, or methods available. Agile is so much more than a single methodology—in fact, you could argue that "agile methodology" in and of itself doesn't exist at all.

An overwhelming majority of project professionals say that their organization has implemented hybrid project management methodologies, many of them combining agile principles with other methods like the more conventional Waterfall method.

So, how do you understand the difference between different agile management approaches, and how do you decide which agile process you should follow?

In this article, I look at those agile frameworks that are beneficial for the project manager who is working in the digital transformation world. This article will help you, as a project manager, to understand that there are many agile flavors, and help you choose which way of working suits your project.

This means I will not go into lesser-known, obscure approaches, or frameworks tied to engineering, program, portfolio, or culture-related frameworks.

First, What is Agile?

Generally, agile is a way of working that prioritizes people, team members, and teamwork over the use of specific processes, and prioritizes results over sticking to specific deliverables or contracts.

It also prioritizes flexibility and creating work iteratively and in increments, rather than segmented and rigid phases (a characteristic of waterfall—read more about agile vs waterfall).

Project planning happens continuously throughout the project, rather than all at once at the beginning. The goal is to deliver working software at each iteration or development cycle. Both software development teams and project teams more generally might choose to work this way.

Read more about agile project management here.

The Benefits Of Agile

Some of the major benefits of agile ways of working include:

- It creates engagement between clients and end users, and increases customer satisfaction
- It can often support culture changes within organizations
- It provides more flexibility, which allows for more project control and the ability to pivot to changing customer needs or business needs
- It reduces waste in the form of meetings and activities that waste time and don't offer value to the project or end product
- It supports faster detection of bugs and other issues, meaning a quicker turnaround time when it comes to fixing them
- It allows for more accountability and diversity of ideas

41 Agile Methodologies, Frameworks, & Approaches

In this article, I touch on the following methodologies, frameworks, and approaches, all of which are rooted in the agile principles.

1. **Scrum**
2. **Kanban**
3. **Scrumban**
4. **DevOps/(Bus)DevOps**
5. **Design Thinking**
6. **Agile Project Management (AgilePM)**
7. **PRINCE2 Agile**
8. **PMI-Agile Certified Professional (PMI-ACP)**
9. **Project Half Double**
10. Agile Program Management (AgilePgM)
11. **Scale Agile Framework (SAFe)**
12. **Large-Scale Scrum (LeSS)**
13. **Nexus**
14. **Scrum at Scale (S@S)**
15. **Spotify Model**
16. **Scaled Agile Lean Development (ScALeD)**
17. AgileSHIFT
18. Agile Fluency
19. Open Space Agility (OSA)
20. Agility Scales
21. Holocracy

22. Sociocracy
23. Disciplined Agile (DA)
24. Praxis
25. Toyota Production System (TPS)
26. Agile Digital Services (AgileDS)
27. Management of Portfolios (MoP)
28. Standard for Portfolio Management (SfPfM)
29. Agile Portfolio Management (AgilePfM)
30. Evidence-Based Portfolio Management (E-B PfM)
31. Bimodal Portfolio Management (Bimodal PfM)
32. eXtreme Programming (XP)
33. Acceptance Test Driven Development (ATDD)
34. Test-driven development (TDD)
35. Behavior-Driven Development (BDD)
36. Feature-Driven Development (FDD)
37. Experiment-Driven Development (EDD)
38. User Experience Design (UX Design)
39. Agile Business Analysis (AgileBA)
40. **Continuous Integration/Continuous Deployment (CI/CD)**
41. Agile Modeling (AM)

As I said, I don't go into depth on the approaches that are geared towards the portfolio-level, engineering-level, or organizational culture. So, I've put in **bold** the methodologies that I do cover in some depth in this article. For others, you can find more detail on my blog.

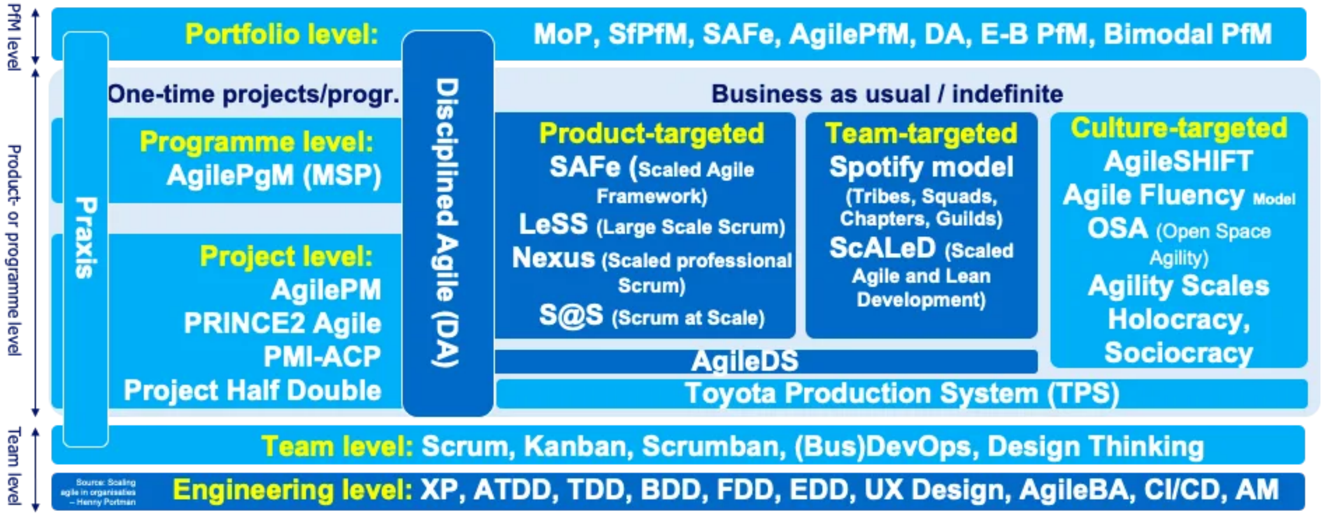
A Bird's Eye View Of The Agile Forest

To get a first impression of the different approaches, I tried to bring some structure into the jungle of approaches, methods, and frameworks.

In the image below, which I call my "*bird's eye view on the agile forest*", I position the 41 best-known agile approaches in a structure (some are in more than one spot). This picture is based on a simpler version in the book *Scaling Agile In Organisaties* that I published in 2017.

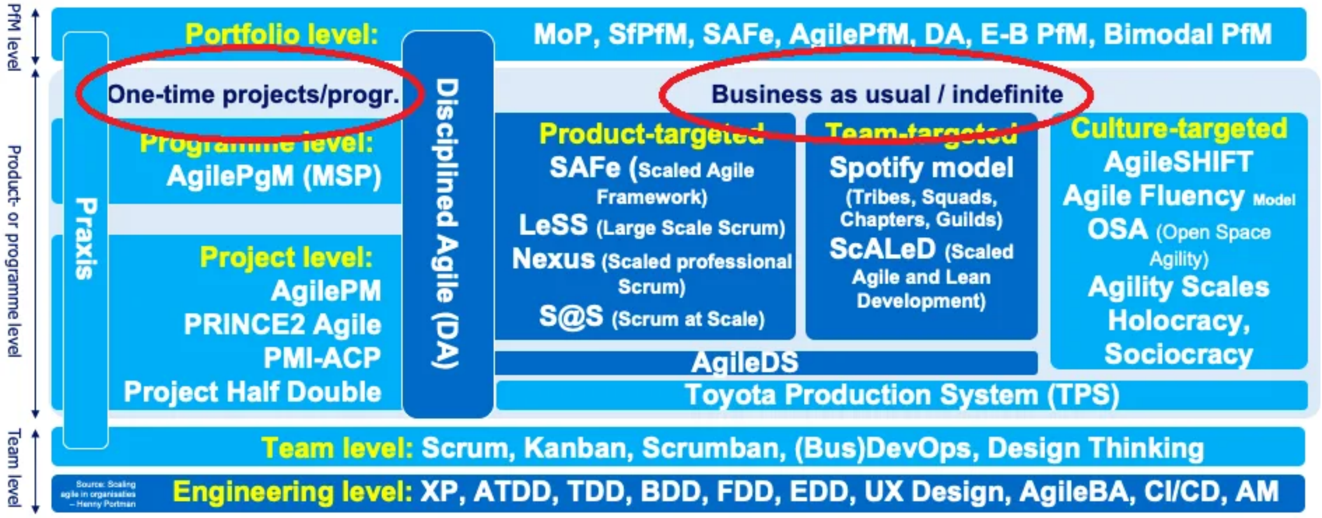
Below, I break down the structure of the picture.

In the dark blue boxes, we see agile approaches that are only applicable in IT-focused organizations. All other approaches are in light blue, meaning that they can be used within IT and non-IT-oriented organizations.


PfM level
Product- or programme level
Team level
Portfolio level:
MoP, SfPfM, SAFe, AgilePfM, DA, E-B PfM, Bimodal PfM
One-time projects/progr.
Business as usual / indefinite
Praxis
Disciplined Agile (DA)
Programme level:
AgilePgM (MSP)
Project level:
AgilePM
PRINCE2 Agile
PMI-ACP
Project Half Double
Product-targeted
SAFe (Scaled Agile Framework)
LeSS (Large Scale Scrum)
Nexus (Scaled professional Scrum)
S@S (Scrum at Scale)
Team-targeted
Spotify model
(Tribes, Squads, Chapters, Guilds)
ScALeD (Scaled Agile and Lean Development)
Culture-targeted
AgileSHIFT
Agile Fluency Model
OSA (Open Space Agility)
Agility Scales
Holocracy,
Sociocracy
AgileDS
Toyota Production System (TPS)
Team level: Scrum, Kanban, Scrumban, (Bus)DevOps, Design Thinking
Source: Scaling agile in organisations – Henny Portman
Engineering level: XP, ATDD, TDD, BDD, FDD, EDD, UX Design, AgileBA, CI/CD, AM

Overview of agile approaches, frameworks, and methods.

The approaches, frameworks, or methods are positioned within two main sections: the “one-time programs/projects” section and the “business as usual/indefinite” section. Some fit within both, so they span the entire area.


PfM level
Product- or programme level
Team level
Portfolio level:
MoP, SfPfM, SAFe, AgilePfM, DA, E-B PfM, Bimodal PfM
One-time projects/progr.
Business as usual / indefinite
Praxis
Disciplined Agile (DA)
Programme level:
AgilePgM (MSP)
Project level:
AgilePM
PRINCE2 Agile
PMI-ACP
Project Half Double
Product-targeted
SAFe (Scaled Agile Framework)
LeSS (Large Scale Scrum)
Nexus (Scaled professional Scrum)
S@S (Scrum at Scale)
Team-targeted
Spotify model
(Tribes, Squads, Chapters, Guilds)
ScALeD (Scaled Agile and Lean Development)
Culture-targeted
AgileSHIFT
Agile Fluency Model
OSA (Open Space Agility)
Agility Scales
Holocracy,
Sociocracy
AgileDS
Toyota Production System (TPS)
Team level: Scrum, Kanban, Scrumban, (Bus)DevOps, Design Thinking
Source: Scaling agile in organisations – Henny Portman
Engineering level: XP, ATDD, TDD, BDD, FDD, EDD, UX Design, AgileBA, CI/CD, AM

I’ve organized agile approaches based on whether they are used for one-time projects and programs, or indefinitely for business as usual.

Then, the approaches, frameworks, or methods are clustered based on which level they generally operate upon: engineering, team, programme or portfolio level.

Even though it’s light blue, the team level is really applicable in both IT-oriented and non-IT-oriented product development, services development, and operations. Meanwhile, the engineering level focuses specifically on IT-oriented product development.

I also cluster some methods based upon their target: are they product-targeted, team-targeted, or culture-targeted? The business as usual or indefinite umbrella frameworks that are used more permanently (both product-targeted and team-targeted) focus specifically on IT and product development. The culture-targeted approaches help organizations increase their agility.

A few caveats:

1. **I haven’t mapped all the known approaches, frameworks and methods** in this figure. And, to be honest, I think there is a lot of duplication, and it’s likely that commercial drivers play a role, too, to “develop” the next kid on the block without added value in comparison with the existing approaches, frameworks or methods.
2. **The line between IT and non-IT applications is not set in stone**. For instance, the one-time, temporary projects and programme frameworks and methods are suitable for both IT and non-IT.

Agile Methodologies That Digital Project Managers Should Know

In this section, I’ll give an overview of some of the most important methods and approaches that digital project managers should know. All mentioned frameworks or approaches embrace the Agile Manifesto and use some form of Scrum, but they differ depending on factors such as whether they are team- or product focused, what level they’re applied on, and more.

Team-Level Agile

Scrum

When teams start working with Agile, Scrum is often chosen. This is an obvious choice, but the question is whether this is always the right choice. In this Roman Pichler blog post, the link was made with the life phase of a product.

During the first phase of a commercial product lifecycle, in which the commercial product is finally put on the market for the first time, the uncertainty is high, and the focus is on on-time delivery of the first market-ready product. A deadline has been set and that date must be met.

During this phase, the focus of the entire development team is on delivering a commercially marketable product. The Scrum Master helps by removing blockers for the team, streamlining their processes, and championing the team and the project, without organizing the team itself. Scrum teams should be self-organizing.

This development process is perfect for Scrum with its iterative approach, being able to deal with uncertainty and working together on the result (the commercial product). Optionally, a second launch can take place with a next set of important functionalities, so that eventually a mature product is put on the market.

Scrum is centered around 5 main ceremonies, which are as follows:

Each ceremony is timeboxed according to the length of the sprint itself. Many of these ceremonies have been adopted (wholly or in part) by teams running other agile software development methodologies, to form a patchwork or approximation of Scrum. Read about the differences between Scrum and Kanban (covered below).

Kanban

In addition to Scrum, on the team level, you will see frameworks such as Kanban (as described in the Kanban Guide for Scrum Teams), or its relatives Scrumban, DevOps, and BusDevOps. This can be used within both IT environments and non-IT environments.

During the course of the product lifecycle, we see the amount of uncertainty and requested changes decrease. At this moment you can make good use of Kanban. In a continuous flow, user stories can be picked up, developed, and deployed one by one by individual team members.

In case there is only one permanent agile team to develop and maintain a product or service, and the team uses Scrum or Kanban with a product owner who prioritizes the product backlog, you might ask if there is a need for a project manager. I would say no, leave that team alone.

Get more info on using Kanban for project management here, and more about the differences between Kanban and agile here.

DevOps, (Bus)DevOps, and CI/CD

If one looks at the often difficult transfer to production environments, the time-to-market can be shortened by properly arranging the transfer and reducing the number of transfer errors when development and production teams are merged, and the integration testing and deployment are automated (Continuous Integration and Continuous Delivery, CI/CD). In this way a DevOps team is created.

Scrumban

Scrumban is the combination of Scrum and Kanban (also known as a hybrid project methodology). In the first instance it was intended as a transitional model to switch from Scrum to Kanban and let the team experience Lean and Kanban concepts.

Nowadays it is an approach in which the team has chosen to work according to Scrum with sprints, but to use the Kanban board and system to continually view and improve its working method to optimize the flow of units of work (ex. user stories).

Scaling Up Towards Product Or Program Level Agile

In order to be able to use an agile way of working in an organization of some size, just having individual agile teams is in most organizations not enough. There are examples of organizations who have or are in the middle of creating a loosely coupled architecture based on micro services (ex. Bol.com in the Netherlands). Each autonomous agile team manages one or more micro services.

However, in most of the organizations the agile way of working needs to be scaled up, and where possible the overarching alignment needs to be taken care of. This can be done by a project manager or by institutionalizing the coordination.

When institutionalizing the coordination, the project manager disappears but many project management tasks are fulfilled by others (ex. the integration team in Nexus, the Release Train Engineer and the Product Manager in SAFe, etc).

To institutionalize coordination, management of dependencies, and integration between the different permanent agile teams within the "business-as-usual" side, there are various frameworks available.

Nexus

Nexus, as described in The Nexus Guide, is a framework for product or software development initiatives with three to nine Scrum Teams, in Sprints of up to thirty days. Nexus is the answer of Ken Schwaber, one of the founding fathers of Scrum, to the scalability of Scrum.

It requires more than just the will and the agile behavior of the different Scrum teams to work together to deliver an integrated product. Nexus is based on and builds on Scrum and the rules and roles formulated in the Scrum guide. We can position Nexus over the team and program levels of SAFe, but it does not offer provisions on portfolio level.

Scrum at Scale (S@S)

Scrum at Scale (S@S, developed by Jeff Sutherland and Alex Brown) is a modular framework. The starting point at S@S is that an all-encompassing one-size-fits-all framework is not possible, but that every time we have to look at scaling the underlying Scrum principles.

The framework can be tailored for your own organization by adding the needed S@S modules. S@S builds on the well-known Scrum framework. By analogy with Nexus you could therefore say that S@S is the answer from Jeff Sutherland, next to Ken Schwaber, the other founding father of Scrum, on the scalability of Scrum.

Large-Scale Scrum (LeSS)

Large-Scale Scrum (LeSS, developed by Craig Larman and Bas Vodde) is an agile framework with rules, based on principles and doing experiments.

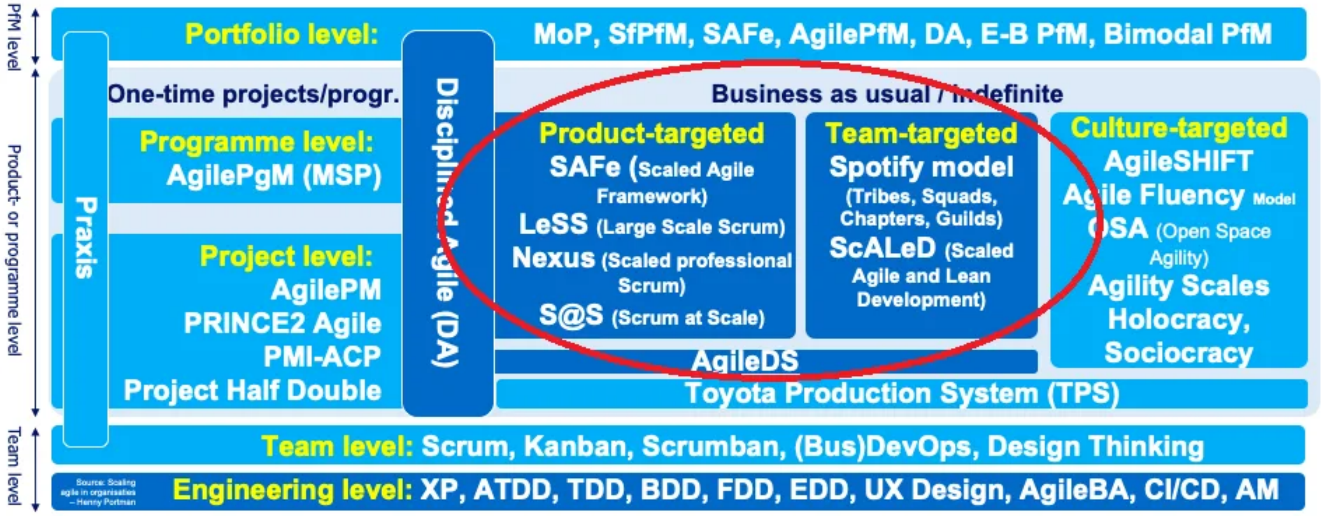
The LeSS company offers a freely accessible knowledge base (less.works) containing the integrated approach, principles, process descriptions, definitions, roles, examples, et cetera, for large-scale, mainly IT-related, product development. Transparency is also a key concept within LeSS. The first version dates from 2005 and since then, work is constantly being done on the use and further development of LeSS.

Scaled Agile Framework (SAFe)

Scaled Agile Framework (SAFe, developed by Dean Leaffingwell) is a framework to enable the scaling up of agile teams in order to create better systems, create higher employee engagement, and make use of correct cost considerations.

This is the mission of the scaled agile organization and of the founder of SAFe, Dean Leffingwell. The scaled agile organization offers a knowledge base that is freely accessible to everyone, with an integrated approach in the form of process descriptions, definitions, roles, examples, etc. for Lean and Agile product development.

SAFe is based on five core competencies: lean-agile leadership, team and technical agility, DevOps and release on demands, business solutions and lean systems and lean portfolio management.


PfM level
Product- or programme level
Team level
Portfolio level: MoP, SfPfM, SAFe, AgilePfM, DA, E-B PfM, Bimodal PfM
One-time projects/progr.
Business as usual / indefinite
Praxis
Programme level:
AgilePgM (MSP)
Project level:
AgilePM
PRINCE2 Agile
PMI-ACP
Project Half Double
Disciplined Agile (DA)
Product-targeted
SAFe (Scaled Agile Framework)
LeSS (Large Scale Scrum)
Nexus (Scaled professional Scrum)
S@S (Scrum at Scale)
Team-targeted
Spotify model
(Tribes, Squads, Chapters, Guilds)
ScALeD (Scaled Agile and Lean Development)
Culture-targeted
AgileSHIFT
Agile Fluency Model
OSA (Open Space Agility)
Agility Scales
Holocracy,
Sociocracy
AgileDS
Toyota Production System (TPS)
Team level: Scrum, Kanban, Scrumban, (Bus)DevOps, Design Thinking
Source: Scaling agile in organisaties – Henny Portman
Engineering level: XP, ATDD, TDD, BDD, FDD, EDD, UX Design, AgileBA, CI/CD, AM

SAFe, LeSS, Nexus, and S@S are in the product-targeted section. They all relate to examples where multiple teams work on a single complex product or value stream.

Looking back at my map of the methods, in the “Business as usual/indefinite” section, we can see the above methods of SAFe, LeSS, Nexus, and S@S in the product-targeted section. They all relate to examples where multiple teams work on a single complex product or value stream (product-targeted frameworks).

There’s a division between product and team targets, namely on the basis of cooperation between teams. In other words, can the individual teams work autonomously (with a team focus)? Or, do they have to work together to deliver a new or modified product (with a product focus)?

Additionally, several approaches (not listed in the figure) make a distinction between products that need cooperation between a maximum of nine teams (in total the team of teams must not exceed the Dunbar number of 125-150 people) and a team of teams of teams (ex. SAFe large solutions, Nexus+, LeSS Huge). An example of this in the real world occurs at Philips, a Dutch medical equipment supplier who uses SAFe and works with 20-30 teams on a single product.

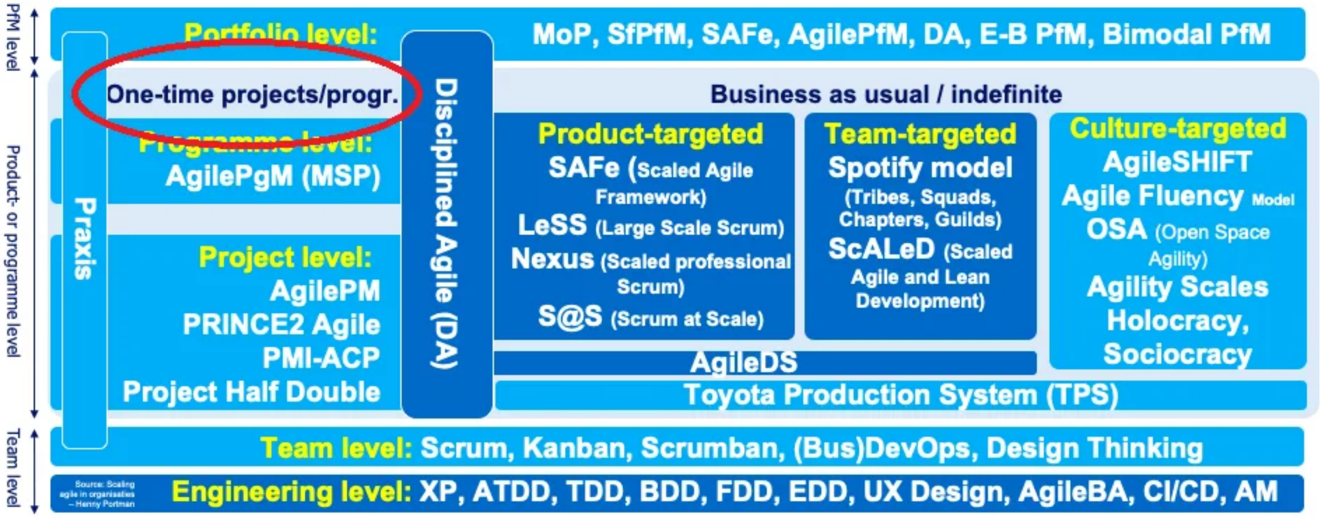
The Spotify Model and ScALeD

Looking to the product-targeted frameworks, we see approaches that support IT departments maintaining dozens or hundreds of applications or services. In these settings, the dependencies between the teams are minimal (multiple team targeted frameworks).

Here, the Spotify model (developed by Henrik Kniberg, Anders Ivarsson, and Joakim Sundén) can be positioned, but also Scaled Agile Lean Development (ScALeD, developed by Peter Beck, Markus

Gartner, Christoph Mathis, Stefan Roock, and Andreas Schliep).

For both groups, there are essential interfaces between the teams in areas such as data integrity, security, and architecture that may not (but sometimes will) ask for coordination when implementing changes. In this group I don't expect to find a project manager. Because these teams are autonomous, I believe that a project manager will not add value.


PfM level
Product- or programme level
Team level
Portfolio level:
MoP, SfPfM, SAFe, AgilePfM, DA, E-B PfM, Bimodal PfM
One-time projects/progr.
Business as usual / indefinite
Programme level:
AgilePgM (MSP)
Praxis
Project level:
AgilePM
PRINCE2 Agile
PMI-ACP
Project Half Double
Disciplined Agile (DA)
Product-targeted
SAFe (Scaled Agile Framework)
LeSS (Large Scale Scrum)
Nexus (Scaled professional Scrum)
S@S (Scrum at Scale)
Team-targeted
Spotify model
(Tribes, Squads, Chapters, Guilds)
ScALeD (Scaled Agile and Lean Development)
Culture-targeted
AgileSHIFT
Agile Fluency Model
OSA (Open Space Agility)
Agility Scales
Holocracy,
Sociocracy
AgileDS
Toyota Production System (TPS)
Team level: Scrum, Kanban, Scrumban, (Bus)DevOps, Design Thinking
Source: Scaling agile in organisations – Henny Portman
Engineering level: XP, ATDD, TDD, BDD, FDD, EDD, UX Design, AgileBA, CI/CD, AM

One-time projects are where project managers are most needed.

On the left side of the map, we see the one-time project frameworks. These are frameworks where the role of project manager is needed. Most of them are a further development of the more traditional project management frameworks:

Agile Project Management (AgilePM)

Agile Project Management (AgilePM, which is derived from Dynamic Systems Development Method, aka DSDM) comes from Agile Business Consortium. It's a method by which different, possibly permanent, agile teams and non-agile teams are coordinated for the duration of a project.

The AgilePM method for managing agile projects consists of a framework, comprising a philosophy, derived principles, and four building blocks: people, workflows, products, and applications.

The AgilePM philosophy is that every project must be aligned to clearly defined business goals and must focus on the early delivery of products that provide real added value to the business organization.

PRINCE2 Agile

PRINCE2 Agile (derived from PRINCE2 from AXELOS) includes both the existing PRINCE2 and the agile way of thinking. The agile way of thinking must be seen as agile behavior, concepts, frameworks, focus areas, and techniques. The existing PRINCE2 principles, processes, and themes remain, but should be tailored using the agile way of working and the project itself.

PRINCE2 Agile searches for the best of both worlds where the emphasis lies in the use of PRINCE2 within project direction and project management, and an agile approach in the product delivery. Depending on the project situation you can apply more or less of the PRINCE2 or agile way of thinking.

Here’s a short video I created about PRINCE2 Agile.

PMI-ACP

PMI Agile Certified Practitioner or PMI-ACP (in addition to the PMBoK Guide of PMI) is not a stand-alone framework, but it is a certification based on different books (and the frameworks and underlying techniques described therein).

Within PMI-ACP, seven domains are identified, each of which is subdivided into a number of task areas. The domains are Agile principles and framework, value-driven delivery, stakeholder management, team performance, adaptive planning, problem detection and solutioning, and continuous improvement.

Project Half Double

Project Half Double is run by a community of dedicated project management practitioners who are passionate about what they do. It was co-created in an iterative way by a community of dedicated project management practitioners.

The methodology is based on the four building blocks to achieve double the impact in half the time: impact, flow, leadership, and local translation.

Impact is all about stakeholder satisfaction. Flow represents high intensity and frequent interaction in project work, learning, and impact. Leadership shows that leaders must embrace uncertainty and make the project happen. Local translation is the fact that you have to tailor the methodology to the needs of the organization.

Disciplined Agile (DA)

Disciplined Agile (DA) covers both one-time projects and programs as well as business as usual product development. The DA toolkit is a process decision toolkit that describes how agile software development, DevOps, IT, and business teams work in enterprise-class settings.

Disciplined Agile (DA) offers a portfolio process in which, in addition to projects, a number of “business-as-usual” aspects are taken into account, such as the permanent teams and the operational management of existing IT solutions.

Agile Digital Services (AgileDS)

Agile Digital Services (AgileDS) is there for delivery and the continuous operations, support, and maintenance of that service (permanent agile delivery team, using the product/service lifecycle).

Learn More About Agile Methods, Frameworks, & Approaches

There are already more than 50 agile approaches, frameworks, and methods and it’s still growing. The map in this post can help you in your agile approach selection process. You can find many more explanations of agile methodologies on my blog, especially of the engineering-related portfolio-level agile methodologies that I didn’t cover in-depth here.

However, it cannot be said enough: Do not act dogmatically, and see an agile practice, approach, framework, or method not as a panacea that can be implemented out of the box.

Common sense helps, too, in achieving greater agility. In fact, the best route to becoming more agile is probably a simple approach that involves dividing your products and services into smaller autonomous parts and having them supported by an individual team.

For more on agile project management, subscribe to The Digital Project Manager's newsletter.

By Henny Portman

Henny Portman is partner of HWP Consulting. He has 40 years of experience in the project management domain. He was the thought leader within NN Group of the PMO domain and responsible for the introduction and application of the PMO methodologies (portfolio, program and project management) across Europe and Asia. He trains, coaches and directs (senior) program, project, and portfolio managers and project sponsors and built several professional (PM(O) communities. He is an accredited P3O, PRINCE2, MSP, MoP, PRINCE2 Agile, AgilePM, and AgileSHIFT trainer, and a SPC4 SAFe consultant and trainer too. He is a P3M3 trainer and assessor and PMO Value Ring Certified Consultant (PMO Global Alliance). In addition, Henny is international speaker and author of many articles and books in the PM(O) field.
Follow the author: Opens new window Opens new window Opens new window