A birds eye view on the agile frameworks forest

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 frameworks available. To get a first impression of the different frameworks, I try to bring some structure in the jungle to methods and frameworks. In Figure 1, I position the best-known agile frameworks in a structure. The frameworks are positioned within the 'One-time programs / projects' sections or within 'Business as usual' / indefinite, or both.

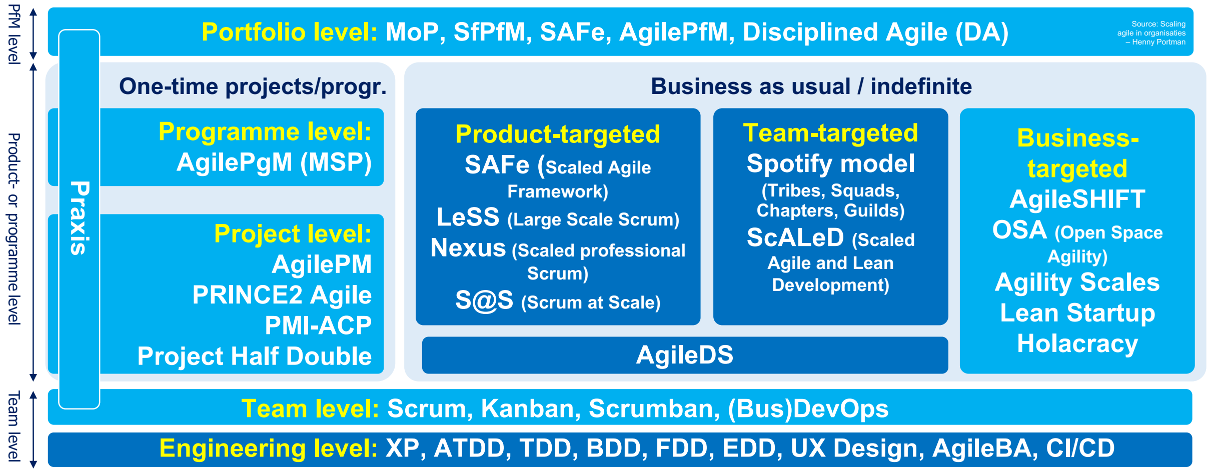
PfM level
Product- or programme level
Team level
Portfolio level: MoP, SfPfM, SAFe, AgilePfM, Disciplined Agile (DA)
Source: Scaling agile in organisaties – Henny Portman
Praxis
One-time projects/progr.
Programme level: AgilePgM (MSP)
Project level: AgilePM PRINCE2 Agile PMI-ACP Project Half Double
Business as usual / indefinite
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)
Business-targeted AgileSHIFT OSA (Open Space Agility) Agility Scales Lean Startup Holacracy
AgileDS
Team level: Scrum, Kanban, Scrumban, (Bus)DevOps
Engineering level: XP, ATDD, TDD, BDD, FDD, EDD, UX Design, AgileBA, CI/CD

Fig. 1 Overview agile framework[1]

On the other side the frameworks are clustered around team, product or programme and portfolio level. In the dark blue boxes in Figure 1 we see agile frameworks that are only applicable in IT-focused organizations. All other frameworks can be used within IT and non-IT-oriented organizations (light blue coloured). I haven't mapped all the known frameworks in this figure, and to be honest, I think there is a lot of duplication and probably commercial drivers play a role too to 'develop' the next kid on the block without added value in comparison with the existing frameworks.

The team level, including Scrum and Kanban, is applicable in both IT-oriented and non-IT-oriented products and services development and operations. The engineering level focuses specifically on IT-oriented product development. The one-time, temporary projects and programme frameworks are suitable for both IT and non-IT. The permanent umbrella frameworks (both product-targeted and team-targeted) focus specifically on IT and product development and the business-targeted frameworks help organisations to increase their agility.

Team level

If we start at the team level in Figure 1, then we see of course **Scrum** as described by Ken Schwaber and Jeff Sutherland in their Scrum Guide. In addition, you will see frameworks such as **Kanban** (as described in the Kanban Guide for Scrum Teams), Scrumban and DevOps or BusDevOps. The team level can be used both within the IT environment and the non-IT environment. At this team level we can position the following IT frameworks too: Crystal Development Framework, Agile Unified Process (AUP) as a simplified version of Rational Unified Process (RUP) which was superseded by Disciplined Agile Development (DAD) which was superseded by Disciplined Agile (DA). If you want to deliver quality as a team within the IT world, only following these frameworks is not enough. To improve quality and minimize technical debt (e.g., inefficient code due to many iterative adjustments), you could make use of eXtreme Programming (XP) with Pair Programming, Acceptance Test Driven Development (ATDD), Test Driven Development (TDD), Behaviour Driven Development (BDD), Feature Driven Development (FDD), Example Driven development (EDD), User Experience (UX) Design, Continuous Integration and Continuous Deployment. AgileBA delivers the techniques to perform business analysis.

[1] This picture is based on a simpler version in the book Scaling Agile in organizaties (Portman, 2017)

Scrum or Kanban?
When teams start working with Agile, Scrum is often chosen. An obvious choice, but the question is whether this is always the right choice. In a Roman Pichler[2] blog 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 team is on delivering a commercially marketable product. This development 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. During the further 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.

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 Deployment CI/CD). In this way a **DevOps** team is created.

Scrumban is the combination of Scrum and Kanban. 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 system to continually view and improve its working method to optimize the flow of units of work (e.g. User Stories).

Scaling up towards product- or program level

In order to be able to use an agile way of working in an organization of some size, just having individual agile teams is not enough. The agile way of working needs to be scaled up and where possible the overarching alignment needs to be institutionalized.

To institutionalize coordination, management of dependencies and integration between the different permanent agile teams within 'the run-the-business' / 'business-as-usual' side there are various frameworks available, including:

- **Nexus**, as described in The Nexus Guide, is a framework for developing 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 behaviour of the different Scrum Teams to work together to deliver an integrated product. Nexus is based 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**, 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 of 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**, 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**, developed by Dean Leaffingwell) is a framework to enable 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,

[2] Pichler, Roman, 'Is Scrum right for your product?', 19 september 2016, see: www.romanpichler.com

Dean Leffingwell. The scaled agile organization offers a knowledge base that is freely accessible to everyone (www.scaledagileframework.com) with an integrated approach in the form of process descriptions, definitions, roles, examples, etc. for Lean / Agile product development. SAFe is based on five core competences: Lean-Agile Leadership, Team and Technical Agility, DevOps and Release on Demands, Business Solutions and Lean Systems and Lean Portfolio management.

Figure 1 (see the 'Business as usual' / indefinite block), makes use of a division between product and team targets, namely on the basis of cooperation, if necessary, of teams or not. Or with other words, can the individual teams work autonomous (team focus) or do they have to work together to deliver a new or modified product (product focus). The fore mentioned frameworks all relate to examples where multiple teams work on a single complex product or value stream (product targeted frameworks). Not visual in the figure several frameworks make a distinction between products where you are working together in with 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 (e.g. SAFe large solutions, Nexus+, LeSS Huge).

The other group concerns frameworks to support IT departments that have to maintain dozens or hundreds of applications or services, whereby 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 or sometimes will ask for coordination when implementing changes.

In addition, there are many, less known, frameworks that can offer support at the product level, including Agile Integration Framework (AIF), Agile Team Portfolio Management (AgileTPM), AgilePath, Continuous Agile, Disciplined Agile (DA), Enterprise Scrum, Enterprise Agility, FAST Agile, RAGE, Surge, XSCALE, Industrial XP, and AgileDS.

On the left side of figure 1 we see the one-time projects and programs as part of ‘change the business’. Here a distinction is made between projects and programs. Within the project block we see three frameworks and/or methods, all three of which are a further development of the more traditional project management frameworks:

- Agile Project Management (**AgilePM**, which is derived from DSDM);
- **PRINCE2 Agile** (derived from PRINCE2 from AXELOS)
- **PMI-ACP** (in addition to the PMBoK Guide of PMI)
- **Project Half Double** (Project Half Double is run by a community of dedicated project management practitioners who are passionate about what they do)
- Agile Project Management (APM), not mentioned in the figure, can be positioned here too.

On the program side we see:

- Managing Successful Programs (**MSP** from AXELOS) that is very agile in itself with the step-by-step growth (via tranches) towards the intended goal (and connects to PRINCE2 (Agile)) and
- **AgilePgM** (Agile Program Management of Agile Business Consortium) that connects with AgilePM on the one hand and is comparable with MSP on the other hand.

Praxis covered the portfolio, programme and team levels. Praxis is a free framework for the management of projects, programmes and portfolios (based on PRINCE2, MSP, MoP, AgilePM and other frameworks). It includes a body of knowledge, methodology, competency framework and capability maturity model. The framework is supported by a knowledgebase of resources and an encyclopaedia.

Portfolio management level

Traditional portfolio management focuses on ‘change the business’. In the previous chapters it has become clear that more and more changes are being handled by the line organization, that is to say: by the permanent agile teams. This means that portfolio management must now also provide an overview of what takes place in ‘run the business’ / ‘business as usual’ for to be implemented change initiatives. Existing portfolio frameworks such as Management or Portfolios (**MoP** from AXELOS) and Standard for Portfolio Management (**SfPfM** from

PMI) only cover the change-the-business part. Agile Portfolio Management (**AgilePfM** from ABC) covers 'run the business' / 'business as usual' as well as 'change the business'.

In addition, there are a number of agile frameworks that also include a portfolio management component:

- **SAFe** offers a portfolio management layer to control 'run the business' / 'business as usual' permanent team(s) of teams.
- Disciplined Agile (**DA**) offers a portfolio process in which, in addition to projects, a number of 'run-the-business' / 'business-as-usual' aspects are taken into account, such as the permanent teams and the operational management of existing IT solutions.
- **Scrum @ Scale** contains modules Strategic vision and Organizational development to which portfolio management can be related.
- Spotify also provides its own portfolio management approach with its strategic planning.
- **AgilePfM** use some basic concepts of an innovation hub, an agile portfolio process, maturity of the initiatives within the portfolio as well as horizons for an agile portfolio.

At the moment (2018) there are no mature portfolio management frameworks that include 'change the business' as well as 'run the business' / 'business as usual'. AgilePfM was launched by the Agile Business Consortium (previously DSDM Consortium) as part of their Agile Business Change Framework. However, it is becoming increasingly clear that the overarching agile portfolio management principles are based on frameworks like SAFe, Agile PfM and Disciplined Agile.

Business level

The business focus provides frameworks to increase business agility by changing the mindset of all staff in the organisation. What does it mean to work in an agile way? How can we make sure that the Agile Manifesto values and principles are understand and applied, and the Scrum values (courage, focus, commitment, respect and openness) are part of what we are doing? If the right mindset is in place it makes it much easier to implement an agile framework. In figure 1 the following frameworks are mentioned:

- Open Space Agility (**OSA**) is a safe, pragmatic and repeatable technique for getting a rapid and lasting Agile adoption. It works with the framework you are currently using, and OSA can be added at any time. OSA is used to actively engage as many employees as possible in your Agile program.
- **AgileSHIFT** (developed by AXELOS) is a framework that prepares people for transformational change by creating a culture of enterprise agility. The AgileSHIFT framework helps organizations to undergo a transformational change, to adopt a 'survive, compete and thrive' mindset. It will help to bridge the gap between the current and the target state (the Delta in AgileSHIFT) by embracing a range of agile, structured and hybrid approaches across the organization. The existing severe split between 'run the business' and 'change the business' will vanish.
- **Agility scales** (developed by Jurgen Appelo) helps organizations achieve agility at scale from the bottom up – with measurable evidence of organizational transformation.
- **Lean Startup** (developed by Eric Ries) is a methodology for developing businesses and products, which aims to shorten product developmentcyclesand rapidly discover if a proposed business model is viable; this is achieved by adopting a combination of business-hypothesis-driven experimentation, by using a minimum viable product (MVP), iterative product releases, and validated learning.
- **Holacracy** (developed by Ternary founder Brian Robertson) is a method of decentralized management and organizational governance, in which authority and decision-making are distributed throughout a holarchy of self-organizing teams rather than being vested in a management hierarchy

Not mentioned in the figure:

- Goal Driven Agile (GDA) rests on three main pillars: autonomy, alignment and structured improvement. It's a very simple framework and consists of only one base structure, the diamond, five roles and ten building blocks.

Already more than 50 agile frameworks and it's still growing. The figure can help you in your agile framework selection process, but it cannot be said often enough, do not act dogmatically, see a framework not as a panacea that can be implemented out of the box. Common sense helps too to achieve more agility and probably the best route to become more agile is dividing your products and services into smaller autonomous parts and have them supported by an individual team.

Henny Portman is blogger / reviewer (hennyportman.wordpress.com), author, international speaker and partner, trainer and senior consultant P3M3 Maturity at HWP Consulting.