Agile Software Development Guide
Agile software development is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.

There are multiple concrete Agile methods. In common, they promote development, teamwork, collaboration, and process adaptability. The two most widely used methods are Scrum and Kanban.

Kanban method can be applied to the software development process. Applying the Kanban methodology helps developers to visualize the workflow, track issues and status changes along the defined and agreed workflow.

Scrum is an iterative and incremental Agile software development framework for managing software projects and product or application development. It defines “a flexible, holistic product development strategy where a development team works as a unit to reach a common goal”.

AGILE: More than Just a Software Development Methodology

To 'be Agile' in business is to be able to react to the unexpected, to be able to adapt to a changing market environment and respond to consumers needs quickly. This also applies to software development.

Agile processes address this lack of flexibility, enabling developers to manage changing requirements. With Agile, developers can produce and release features and then continue to add additional features thereafter. There are many development methods such as Waterfall, Scrum or XP and these methodologies are still all in use today, however they are increasingly being fused together with Agile processes, creating hybrid methodologies, solutions tailored to individual business requirements.

At its heart ALM Software is a Collaboration tool, designed for efficient working processes, tailored for individual business needs. An innovative ALM solution can be also used in non software development implementations such as in marketing. It is built to accommodate SAFe requirements and is therefore fully scalable for enterprise use and it is perfect for geographically dispersed users.

Agile development is built upon development methods of the past. Methods that were, and still are inflexible because the software development lifecycle (SDLC) uses defined phases of development in which the phases require that one phase of development must be completed before the next is begun.
Is Agile better fit for fast growing, small development companies or is it for software giants?

The answer is of course both, but implementation happens differently.

In the case of Spotify it seems that successful implementation of Agile lead to quick and manageable growth. Scrum is mentioned as the key for success in competition with technology giants like Apple, Amazon, and Google who were all using Scrum, but on a far greater scale, with 15,000 or more programmers.

Dr. Jeff Sutherland, the co-creator of Scrum analyzed the main factors of Spotify’s agile success. For start-ups and small companies with 50 - 100 programmers Scrum is fundamentally a better fit than any other development method. “This is the key to competing with new industry standards in software development” – said Dr. Sutherland.

Scrum enables quick delivery of results, with focus on customer values, and sets collaboration and communication as top priorities while encouraging lean thinking. It makes the team environment a happier place by facilitating a transparent, cooperative attitude at work.

In Spotify’s case the key elements of Agile success are highly educated management in Agile practice, where teams are led by masters of Scrum. Furthermore, Spotify kept teams small, working in small clusters of development teams, with each team dedicated to a piece of the development work. This ensures easier change, and upgrade constantly without breaking anything else. In addition, their focus is on constant problem solving, removal of roadblocks. When a business is growing, roadblocks will arise that can slow down the speed and sometimes risk the quality of software development. The correct attitude is to deal with those issues by constantly monitoring potential obstacles, and removing them before they become a bigger problem.
Agile Project Management Tools can be invaluable support for teams working with Scrum. codeBeamer supports SCRUM based development with Kanban board agile planning board and sprint-release planning. In Javaforge small and start-up companies can try and use it for free for Open Source Projects. For enterprise use codeBeamer is available with special pricing conditions for start ups with small teams (up to 10 users). Intland Software is committed to assisting start ups and small enterprises to reach their Agile potential!

Agile ALM: Slash Administration for Software Developers

Software development has always been difficult and risky for business; however the ability to manage risk and avoid potential failures before they arise is greater than ever before.

When working with Waterfall, teams collect product requirements before the development process begins and as a result they often can’t adapt to changed requirements as quickly as they should. The result is that the final product often does not meet expectations. This is one reason why Waterfall is increasingly being replaced by Agile development.

Agile software development enables teams to re-evaluate products at any time in the development process and turn the development in different directions to adapt to changes more quickly to meet requirements.

Agile adoption is accelerating, however it requires ALM vendors to integrate Agile methodologies into the ALM tools and this is how “Agile ALM” was born. Agile ALM is a great solution to adapt to changes more quickly and release products faster. While ALM manages software development life cycle, Agile enables companies to focus on improving the development process.

ALM tools can no longer be seen as optional but rather as “necessary tools” to increase productivity and reduce expense. ALM tools are developed to manage software from multiple different vendors across the enterprise in one single platform that reduces operational expenses and overwhelming administration. The right ALM tool simplifies business compliance of industry regulations by providing transparency and traceability.

Benefits of Agile ALM in a nutshell:
- +20-50% more productivity
- +10% more engagement and better collaboration
- +30-50% improved quality and satisfied customers
- +20-50% faster delivery

Learn more about how codeBeamer ALM software solution supports different software development methodologies.
The Agile Manifesto was introduced in 2001 and since then Agile software development, or let’s call it the “Agile Movement” has received a lot of attention, growing year on year. According to the 7th Annual State of Agile Development Survey, more than 84% of the companies surveyed had adopted Agile for software development. One of the biggest challenges and questions for many organizations is how to measure the success or failure of an Agile approach.

There are many ways to measure Agile performance including:

- Burndown charts
- Velocity over time charts
- Current velocity
- Remaining time and
- The overall break down of the iteration.

**Burndown Chart in a nutshell**

Burndown chart is one of the most well-know charts. It was developed by Ken Schwaber to provide Scrum community with a tool to plan their work. Burndown chart is a very effective planning tool if you know how to interpret the metrics. Let’s look into the basics quickly:

- Burndown chart is a graphical representation of work left to complete over time.
- It shows effort/cost spent and schedule on a daily basis and helps to lower the potential risks associated with, and required by traditional project management.
- It is a great communication tool empowering customers and stakeholders alike, they can follow project progress on a daily basis.
- The ‘remaining task hours’ can be misleading since it can mean different things in different scenarios – i.e. 30 hours not completed or 10 hours completed and 20 hours not completed work hours. You should look into the details to interpret the information correctly.
• If you look at the graph you will see if Sprint commitments were met and if so, how smoothly. You can see if your team stretched at the end of the Sprint to meet commitment as well as if the team’s performance was consistent or not; with other words you can see if your team is on track or not and reallocate or reprioritize work accordingly.

Burndown Chart: Scenarios to Consider

While nothing is black and white, there are scenarios they can cause your burndown chart to be misleading:

• If it takes longer than 12 hours to complete a task, it can be difficult to track. Encourage team members to create tasks shorter than 12 hours.

• Teams usually update the effort column on a daily basis but they need to re-estimate how much effort they will need to complete the task. If they don’t, “effort remaining” and “effort spent” graphs won’t be consistent and can be easily misunderstood. Team members should understand that they need to update “effort remaining” on a daily basis to have a burndown chart that shows accurate, daily status.

Burndown chart is the most popular tracking tool for organization implementing Agile due to its effectiveness and simplicity. However, use it with the aim to enable your team to produce potentially shippable and working increments.

codeBeamer ALM (Application Lifecycle Management) solution provides your team with 5 key modules to manage software development lifecycle as well as diverse charts to measure Agile performance.
The statistics define success, when software development projects are implemented within the given timeframe, on-budget and with the requested features. Does it mean that agile methods are the universal key to success? Is Agile better than Waterfall?

The general answer is very simple: No, – any methodology by itself is not a recipe for success. Projects can be different, in terms of duration, budget, complexity, number of stakeholders. Agile is not a „one size fits all” approach. Waterfall projects suits to some types of projects and Agile to others, usually smaller, less critical, more local scope and more oriented to end-user. Risk of failure is a nature of software development.

Statement: „Agile projects are successful three times more often than non-agile projects, according to the CHAOS report from the Standish Group.“

---

Waterfall

- Successful: 57%
- Challenged: 29%
- Failed: 14%

Agile

- Successful: 49%
- Challenged: 42%
- Failed: 9%

Source: The CHAOS Manifesto, the Standish Group, 1012
Does it than mean that Agile projects simply manage risk better than Waterfall?

The answer is yes, but it does not mean that failure will be avoided by using Agile methods. Agile methods provide a good toolset that helps to minimize and manage risk effectively. Simply because agile requires better collaboration, more communication, end-user involvement, shorter, more iterative phases with quicker delivery of results. The success mostly depends on the stakeholders of the project, – success is the success of a team, not the project itself. The successful implementation of Agile software projects also depends on the maturity of the client’s expectations. A major portion of software project failure is due to a client who is unable to define the scope.

Summary:

- Agile is not 100% recipe for success
- Risk and failure are both nature of software projects
- Methods can be applied according to certain circumstances – case by case decision
- Requirements management is a critical part of success

Using a comprehensive integrated ALM tool which supports both Waterfall and Agile methods gives the freedom to apply methods that fits to the project attributes.

Check how codeBeamer supports Agile and Waterfall.
Project managers often face long development cycles, constantly changing priorities that disrupt the product development process and consequently leading to a final product that does not meet expectations.

Traditional project management methodologies are often blamed for being unable to adapt to the changes which often makes testing products or changing alterations quite impossible. The Agile approach is often considered as the “perfect” solution, especially for projects with multiple changes. Time is usually less critical in the “Agile world” while iteration status can be constantly checked and adjustments can be made as long as the product is not ready to be released. The Agile approach has many positive points, it is important to note that it is not a “one-fits-all” solution and as a result we often see hybrid solutions.

Hybrid models are usually born from the compromise between different methods. One of the most popular today is Agile-Waterfall Hybrid described by Erick Bergmann and Andy Hamilton. The Agile-Waterfall Hybrid model is far from perfect and it is a compromise. It has as many advantages and disadvantages, just as all methodologies. The Agile-Waterfall hybrid it is often considered as a smart approach to adapt both methodologies without compromising too much.

The most important details to know about Agile-Waterfall Hybrid

- Implementing Agile-Waterfall Hybrid allows software teams to work “Agile”, while hardware development teams and product managers can use traditional PMP/waterfall approach.

- Tight, continuous integration between PMP and Agile software development processes from product concept until validation and production. As with all software development methodologies, collaboration is key and Agile-Waterfall Hybrid interface enables teams to define requirements and adapt to changing requirements and provide feedback from both PMP and Agile sides allowing for continuous delivery.

- The hybrid model is best suited for reuse of software code where dealing with a series of similar products and where future products must also be considered. Such situations result in lead to a quick turnaround, with continual product releases. Backlog management is critical area for successful adoption of this hybrid model and adoption is best assisted by software version release planning features.
When companies **go Agile** for the first time they tend to adopt Scrum. For enterprise the key benefit of using Scrum in software development projects is to be flexible enough to manage changing requirements and develop better products—faster. Scrum enables teams to package the development work into short periods called sprints and include the client within the development process.

The Agile Waterfall model aims to retain the dependency tracking and clarity of Waterfall whilst embracing the strengths of Agile methodology, providing the flexibility and transparency necessary to adapt to the fast changing requirements of stakeholders.

As with all hybrid models both sides must compromise, where the Waterfall development must give up some certainty of fixed expectations, for the flexibility and freedoms of the Agile world. The Agile compromise is to be creative but with far less freedom, working against a fixed deadline with cost forecasting and risk assessments.

**Being Agile: Switching from Scrum to Kanban**

![Diagram of Scrum and Kanban symbols]

[92x518] Being Agile: Switching from Scrum to Kanban

[44x128] When companies go Agile for the first time they tend to adopt Scrum. For enterprise the key benefit of using Scrum in software development projects is to be flexible enough to manage changing requirements and develop better products—faster. Scrum enables teams to package the development work into short periods called sprints and include the client within the development process.

[44x770] The Agile Waterfall model aims to retain the dependency tracking and clarity of Waterfall whilst embracing the strengths of Agile methodology, providing the flexibility and transparency necessary to adapt to the fast changing requirements of stakeholders.

As with all hybrid models both sides must compromise, where the Waterfall development must give up some certainty of fixed expectations, for the flexibility and freedoms of the Agile world. The Agile compromise is to be creative but with far less freedom, working against a fixed deadline with cost forecasting and risk assessments.

**CHAPTER 2**

Agile Software Development Methodologies
Teams benefit from constant feedback and it creates a quick turnaround from initial identification of problems and bugs or additional requirements for new features, to final implementation. It increases the rate of production whilst reducing the cost of production and it produces a better featured product.

**Scrum or Kanban: Which One Is Better?**

No question about it that Scrum is a great Agile software development framework, however its performance can be beaten by Kanban due to bottlenecks in work flow for testing and deployment. By using Scrum product owners can determine the user stories and turn these into sprints. Small stories often fail to produce a shippable product by the end of the sprint and as a result, multiple sprints are required and this results in increased complexity that can be difficult to manage. Kanban addresses these weaknesses of Scrum.

**How Does Kanban Improve Decision Making?**

Kanban is all about real-time decision making and visualization of workflow. As a result Kanban is suited to specific areas of business that requires quick decisions, where there is no time to talk endlessly about the consequences.

For a development team working with Kanban, every team member is trusted and empowered to take make quick decisions. Scrum is great to practice for Agile working and team building, whereas Kanban greatly improves on Scrum by assisting developer teams to act more quickly to avoid delayed decisions and unnecessary expenses.

**Quick comparison of Scrum and Kanban**

- Work commitments required by Scrum, are optional in Kanban
- In Scrum work items should be broken down to be completed within 1 sprint as opposed to no size limits in Kanban.
- Recommended use of Burndown Charts in Scrum whereas in Kanban no particular diagram is prescribed.
- Estimation necessary in Scrum whereas in Kanban estimation is optional.

Many of the requirements of working with Scrum are optional in Kanban. This is because Scrum represents an earlier idea of Agile; Kanban removes many of these working requirements of Scrum, and instead are offered as optional extras in Kanban, only to be implemented if the project requires it. The good news is that there is no need to switch entirely from one to the other but rather a partial transition process by process from Scrum to Kanban is possible as the team becomes more experienced. Alternatively, just keep the best of both Scrum and Kanban as a permenant fixture if it works for your business, the combination of the two is commonly referred to as Scrumban.

Check out how to practice Scrum or Kanban with codeBeamer ALM solutions.
The Main Benefits of Using Scrum


Kenny highlighted the main benefits of using Scrum in development, which is speeding up delivery of new products and new features by developing in short sprints. Kenny defined 3 main areas of benefits. The first and most important is getting delighted customers, which means, in Scrum sprints are really short, so results are delivered and ready for test within 1-3 weeks. The Scrum methods main focus is to provide new features or corrections frequently, and collect feedback from clients as quick as possible. Therefore Scrum speeds up bug fixing processes and new feature development, that makes customers happy.

Applying Scrum in development can reduce the cost of production. Product cost is determined by time and effort spent on new releases, but companies are rarely able to estimate real cost of production. In Scrum story pointing can be applied to estimate tasks' complexity. Complexity strongly relates to cost of production. Story pointing on each tasks gives the opportunity to better estimate the cost of developing new features, and also allows project managers to prioritize tasks according to their complexity.

The third big area of Scrum' benefits is provided by the strong focus on collaboration and daily communication. Scrum provides visibility on tasks for each team member, to track progress and allow better allocation of resources. The strong collaboration focus leads to more happy and productive team.

Scrum as an agile method that ensures quicker response to market demand and leads to cost savings via increased performance of teams. According to my experience Scrum can be used not only in software development, but in sales and marketing as well.

“Delivering features today generates revenue and feedback today.”

Read more about Scrum
Why use Kanban in software development?

Kanban, as a method has its roots in production and is an important element of both Kaizen and Lean production. Kanban is focused on pull strategy, with resources allocated based on where and when they are needed to support just in time delivery. But how can Kanban be applied in software development and what are the benefits of using this method?

The principle method of visualization in Kanban is the Kanban board and in a production environment, this is a physical board where tickets for the demand of resources are placed. In software development it is impossible to use physical boards, where teams are working in virtual environment mostly in distributed locations. In software development a virtual Kanban board can be applied. Kanban is a method that greatly assists teams to be Agile and therefore agile project management and ALM tools usually provide a Kanban board (electronic cardboard) view.

How Kanban delivers benefits in software development:

- **Easier changes**
  Cardboard column names represent the different states of workflow. Visualizing the flow of work is core to understanding how work proceeds, and ensures easier decision-making on changes.

- **Shows WIPs**
  The critical element is that work-in-process at each state in the workflow is limited and that new work is “pulled” into the new information discovery activity when there is available capacity within the local WIP limit.

- **Helps to manage the flow**
  The flow of work through each state in the workflow should be monitored, measured and reported. Kanban board is also helpful for project managers to get insight into the progress of work.

- **Supports problem solving and risk management**
  Due to the explicit understanding of how things work and how work is actually done, it is possible to move to a more rational, objective discussion of issues, problems, risks among team mates. Using this method it facilitates the consensus around improvement suggestions.

- **Forces collaboration and experimentation**
  Kanban encourages small continuous, incremental and evolutionary changes that stick. For example Kanban can support effectively the daily stand up meeting of Scrum, as it gives a quickest overview on tasks, status, WIPs for a better resource allocation and increases the motivation and self-governance of teams.
In the beginning agile development was used for pilot projects at enterprise level, developers worked in small teams experimenting with working “lean and agile methods.” The pioneers noted what worked and what didn’t, and developed various frameworks and strategies for scaled agile adoption.

Agile adoption by enterprises has boomed in the last few years. As more enterprises realized its benefits, more executives started to pay more attention to implement and practice lean and agile methods. The most important management responsibility is still to measure improvement and ROI. One vital role of management, is to select the software solution that best fits to their business model, products and assists to shorten the delivery and release phases in order to outperform competitors.

Once enterprises decide to “go agile” they can choose from different methodologies and frameworks. However experts and consultants tend to specialize in one and promote it exclusively, there is no universal agreement about which is the “best framework or method”. There is no one-size-fits-all solution and every framework has pros and cons.

Quick insight into Scaled Agile Framework (SAFe)

Scaled Agile Framework (SAFe) is one of the most implemented scaled agile frameworks of the time. Let’s have a quick insight into SAFe:

- What is Agile and Scrum for teams, that is SAFe (Scaled Agile Framework) for companies
- Scaled Agile Framework (SAFe) is created by Dean Leffingwell
- Describes three levels of scale including team, program and portfolio level
- Describes the working progress between team level development and business strategy
- Provides PSI and Release Planning
- Constantly evolving, free IP framework
- More information and success stories are on the SAFe homepage.

Learn more about Scaled Agile Framework (SAFe)
Agile Adoption

Agile is used widely by software developers’ and Scrum teams for more than a decade. Now Agile which is an alternative to Waterfall seems to be the “new standard” for enterprises. Today more and more enterprises are realizing the benefits of Agile and the implementation of Agile is ramping up; – especially in the industries where it is important to shorten the delivery and release phases. The real challenge is how to adopt teams’ practices where every team member is responsible for implementing and executing certain tasks at the enterprises level.

If enterprises decide to “Go Agile” they can choose from different methodologies and processes such as Scrum, XP (eXtreme Programming), Crystal, FDD (Feature Driven Development), DSDM (Dynamic Systems Development), Adaptive Software Development, RUP (Rational Unified Process) or the more recent Scaled Agile Framework (SAFe) according their needs.

As Ken Schwaber stated SAFe is built on different iterative and incremental predecessors such as Rational Unified Process (RUP), the popular iterative and incremental software development process framework. The key question is if enterprises really benefit from Scaled Agile Framework (SAFe) developed by Dean Leffingwell and which is all about using the Agile framework at the Enterprise level. The Agile Enterprise Adoption with the Scaled Agile Framework is accelerating, and speaking from our own experience, we have seen remarkable results achieved by adopting SAFe at Enterprise Level.

Learn more about our codeBeamer (commercial version) or try out the free open source version of codeBeamer at JavaForge.com.
We started to develop codeBeamer over a decade ago now and it still remains the most innovative product on the market place. With the new release of codeBeamer 7.2 it has become the most comprehensive implementation of SAFe in the Agile world. It sets itself apart from its competitors in that it enables Agile planning / scheduling while maintaining an established discipline of requirements, tests and development management tools. Another key differentiator is that codeBeamer provides Versioning (baselines) for all artifacts for compliancy, so vital for quality assurance (QA) management and traceability. Besides SAFe codeBeamer ALM solution also supports software development methodologies such as Agile, Waterfall and Agile-Waterfall hybrid.

Download codeBeamer 7.3
or contact our Sales Department at sales@intland.com to request a demo.