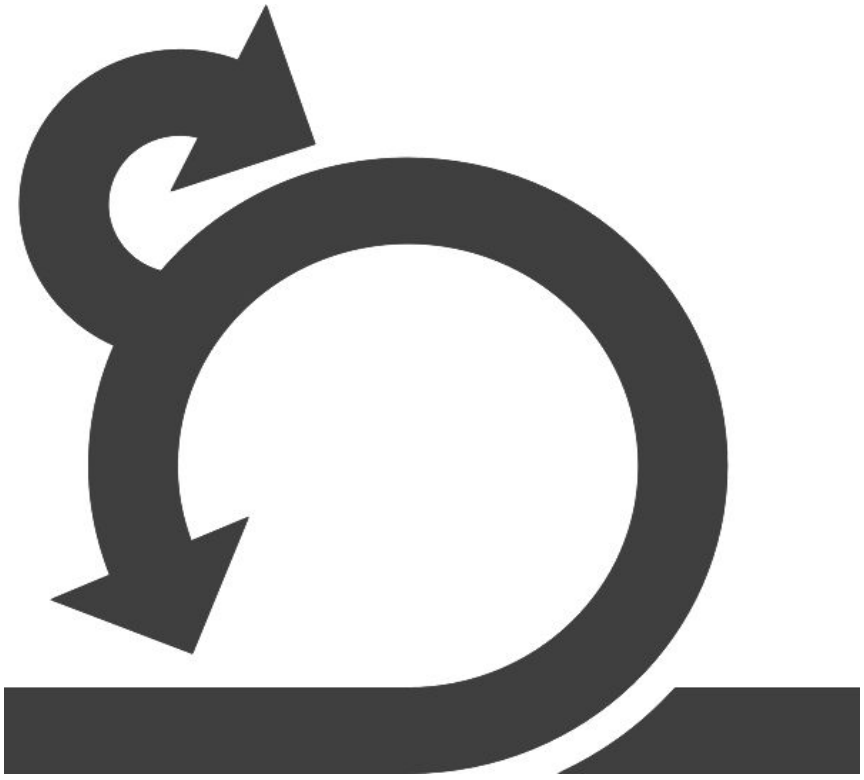


AGILE

Handbook for Senior Management

Project & Portfolio Managers, General Managers, Executives & Founders

CONTINUAL.DELIVERY

Overview

This handbook is meant to be a quick-starter guide to Agile Project Management. It is meant for the following people:

Someone who is looking for a quick overview on what Agile is and why it is awesome.

Someone who needs help getting their head around Agile project management.

Someone who is looking to introduce Agile on their next project.

Someone who needs help selling Agile to their boss or client.

Traditional Project Management

I have structured this book to match the lifecycle of a traditional project. For the initiation and closing phase's you still should create a statement of work, charter documents, lessons learned, budget reviews all depending on your organization's expectations.

1. Initiation
2. Planning
3. Executing
4. Monitoring
5. Closing

The difference is 2-4 phases which in traditional are done once. In Agile they are repeated in iterations enabling feedback loops. As the phases are repeated there are different artefacts and ceremonies to enable this.

Further Resources

This handbook is not meant to be the end-all-be-all to agile. It is meant to give managers an overview of the framework, terminology, and its benefits in 15 minutes or less. The resources section lists recommended books and companies that can provide a more robust training on how to implement it.

The Agile Handbook for Project Managers

Who Am I

Who am I to be writing Agile Handbook for Project Managers?
My name is Shane Drumm and I'm one of the co-founders of Continual.Delivery. We are Marketing Tech Professionals who can provide you with insights on all areas of a Digital Transformation from New Technology, WebOps to Agile Digital Marketing.

Over the years I have worked with numerous project managers who weren't used to working in an Agile environment. They struggled to understand the dynamics and how to project manage it. This leads to confusion, uncertainty, assumptions and ultimately failed projects.

I'm certified with the PMI® as a traditional Project Management Professional PMP® and Agile via the Project Management Institute Agile Certified Practitioner PMI-ACP®. Also, have been a qualified Scrum Master with the Scrum Alliance since 2011. Have taught project management in classroom setting for 2 years and have a project management blog which I wrote all the content myself called pm-training.net.

Most importantly, I have worked in a variety of Agile roles in large companies and have seen first hand what works and doesn't. The majority of these companies were going through Digital Transformations and moving from a project first company to a product. Being a product company working in an Agile is essential as it enables feedback loops and fast delivery.

Hope you find the handbook useful.

Kind Regards,
Shane Drumm

shane@continual.delivery
0490-932-552

https://www.linkedin.com/in/shanedrumm/

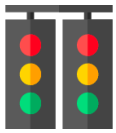
Why Agile

Waterfall is perfect for projects where its a repeatable task such as construction projects or civil work where you are not expecting a lot of changes based on feedback.

Agile started in software development but can be used in any project where requirements are expected to change based on feedback. In Agile projects you work in iterations to constantly gather feedback and alter design as you see fit.

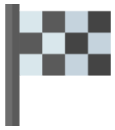
SPEED TO MARKET: Agile enables teams to get a concept to users as quickly as possible to gather feedback.

FLEXIBILITY: Agile expects changes based on feedback from users and follow rolling wave planning approach.



QUALITY: Agile integrates testing throughout the process means less time spent on QAing at the end.

COST CONTROL: Agile is flexible with regard to scope. Agile isn't about paying a lot with uncertainty, it's about paying for only what you need.



RIGHT PRODUCT: Incremental releases let you test your product early and often.

Agile From First Principles

I'm a strong believer in understanding the Agile principles and not work within a strict rules based environment. Agile is all about feedback loops and making small continuous changes to the process to maximize productivity.

To understand the principles though you need to understand the different aspect of Agile first. You should first understand its origins from the agile manifesto and 12 Agile principles both originally published by agilemanifesto.org.

Agile is a development methodology based on iterative and incremental approach. Scrum, Kanban, Extreme Programming and Lean are different implementations of agile methodology.

The Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

The 12 Principles of Agile

Our highest priority is to satisfy the customer through
early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes
harness change for the customer's competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of
months, with a preference to the shorter timescale.

Business people and developers must work together
daily throughout the project.

Build projects around motivated individuals. Give them the environment and
support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and
within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development.
The sponsors, developers, and users should
be able to maintain a constant pace indefinitely.

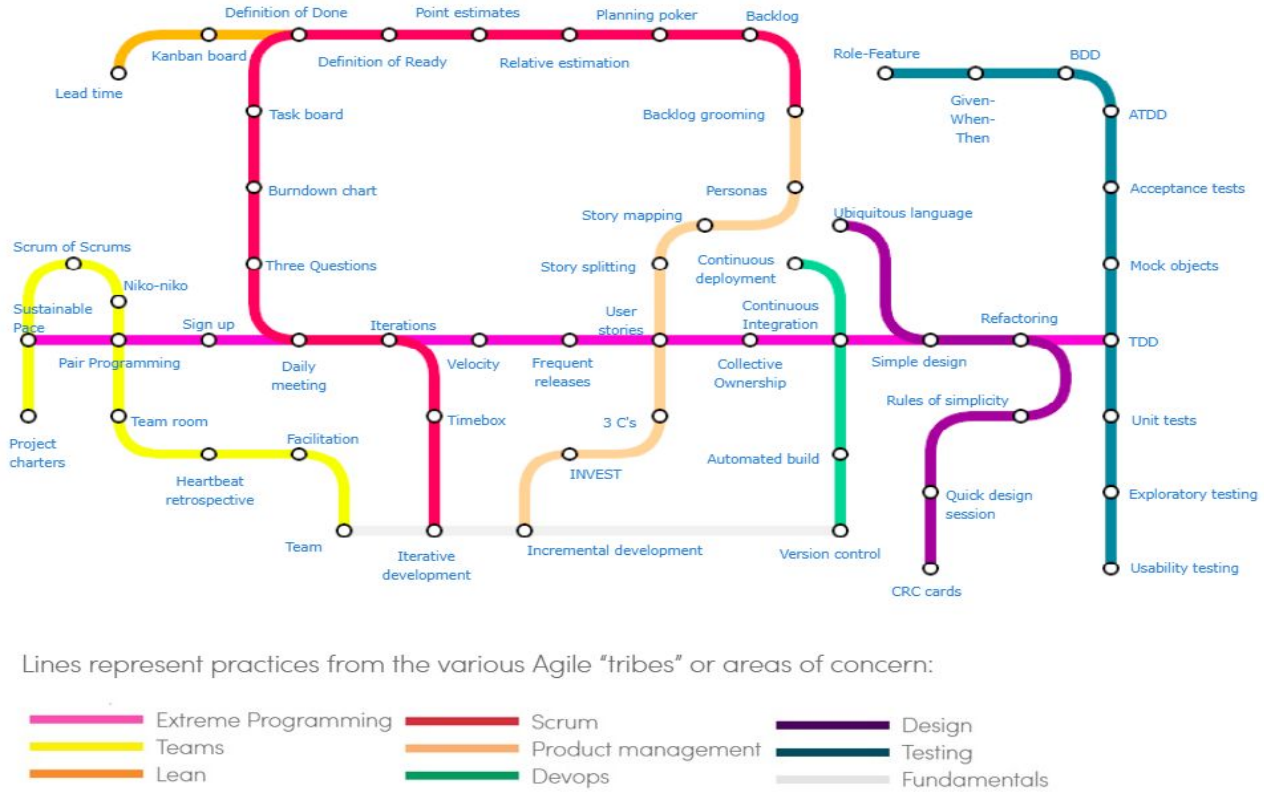
Continuous attention to technical excellence
and good design enhances agility.

Simplicity--the art of maximizing the amount of work not done--is essential.

The best architectures, requirements, and designs
emerge from self-organizing teams.

At regular intervals, the team reflects on how to become more effective,
then tunes and adjusts its behavior accordingly.

Overview of Agile Practices



Subway Map of Agile Practices - Source: Agilealliance.org

The Subway Map of Agile practices does an excellent job of illustrating the overlap of agile methodologies and practices.

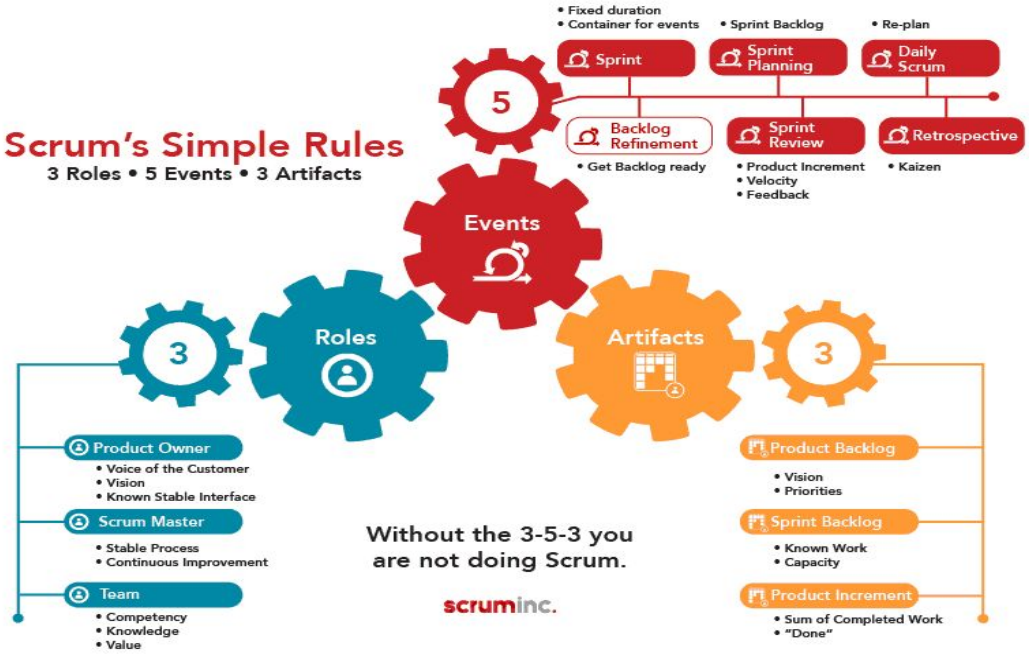
Every organisation and project will have different constraints and flexibility so the process will need to be modelled to suit these needs. Different companies implement different versions of Agile, some stick to one methodology such as XP others hybrids of different methodologies such as Scrum and Kanban.

It is recommended though new teams should use "out-of-the-box" agile implementation and then change after a few iterations. The reason this is recommended so the team knows why the certain process is there, need to recognise the value in the process before deciding not suitable.

Process tailoring is a continuous process when it comes to Agile, we are always trying to improve and make it more efficient by reducing waste and focusing on what works versus what doesn't.

Agile Methodologies

Scrum



Source: https://www.scruminc.com/the-3-5-3-of-scrum/

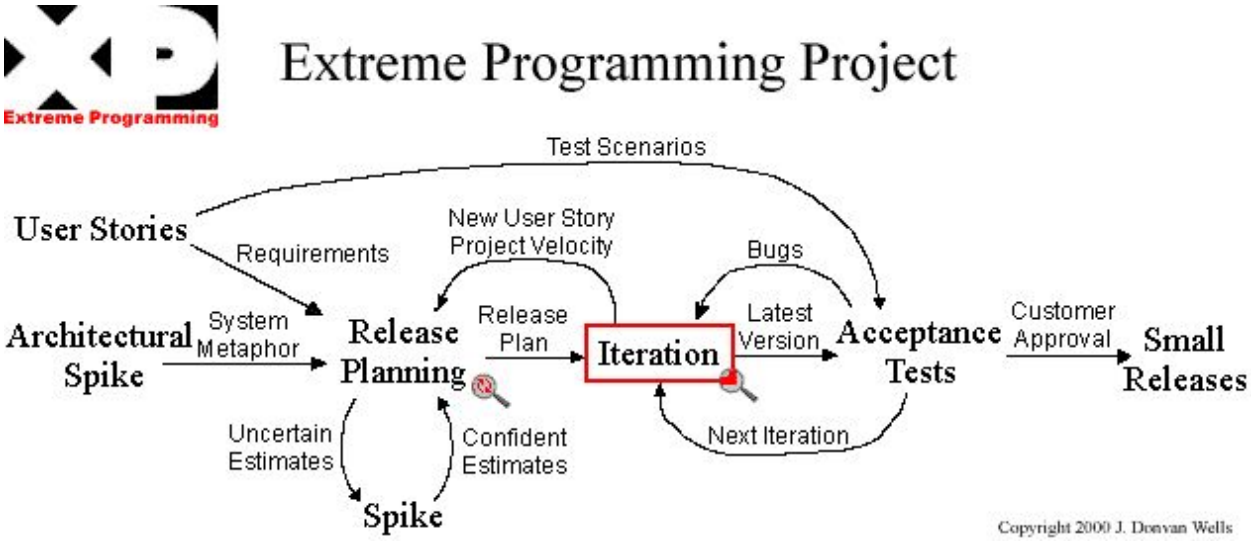
The Scrum development methodology is an iterative and incremental software development methodology. Scrum has a simple set of roles and responsibilities to be followed and has nothing to do with rugby. Scrum is based on prioritizing a backlog and committing to doing a set of work in a short fixed time period and no other work to be done in the space of time. This approach is in direct response to the traditional waterfall method which didn't react to change.

Scrum teams can easily react to new requirements/change as they have only committed to two or three weeks of work. A Scrum team will have a Scrum Master who is there to help the team implement the methodology and maintain the practices.

Further Reading:

- https://www.scrum.org/
- https://www.scrumalliance.org

Extreme Programming



The 5 principles behind Extreme Programming (XP) Development Methodology are:

1. **Simplicity** – Only do what is needed and asked for.
2. **Communication** – Everyone is part of the team and communicate daily
3. **Feedback** – Demonstrate software early and often and listen to any feedback
4. **Courage** – Tell the truth about progress and estimates
5. **Respect** – Everyone contributes value even if it is just enthusiasm

As other Agile development methodologies; XP puts emphasis on delivering quickly and early to gather feedback.

Further Reading:

- http://www.extremeprogramming.org/
- http://agilemodeling.com/essays/agileModelingXP.htm
- http://www.jamesshore.com/Agile-Book/the_xp_team.html

Lean Software Development

Lean manufacturing is the basis for Lean Software Development which is the basis for Kanban. Kanban is a framework for managing flow of materials or information. Kanban matches amount of work to a teams capacity giving them more flexible planning options, faster output and transparency throughout the development cycle.

Kanban has one main tool the Kanban Board, which augments the traditional Iteration Backlog with additional detail by including the development steps/processes as well as introducing work limits per queue.

Even though Kanban doesn't define a full agile life cycle it has gained popularity as it replaces the Iteration Backlog and can adapt to change a lot quicker. It is also been used by companies new to Agile as it can fit in with their current process without much interruption. Kanban works best with:

- Small tasks that don't warrant a story.
- Support team as it prevents overloading team members with work and they can just pick up next tasks on top of the queue.

The core principles of Lean Thinking were created by Mary and Tom Poppendicks is as follows:

1. Eliminate Waste
2. Optimizing the Whole
3. Delivering Fast
4. Building in Quality
5. Respecting People
6. Constant Improvement

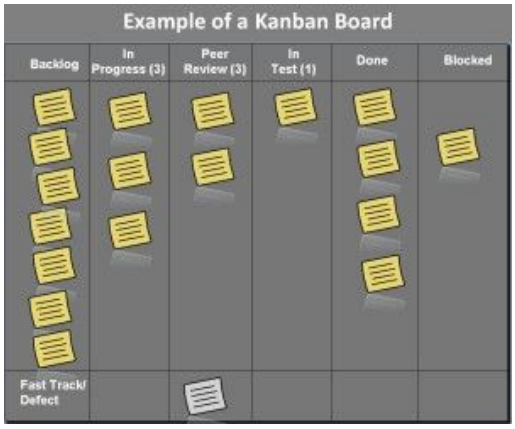
Further Reading:

- http://www.poppendieck.com/

Kanban Development

Rule #1: Visual Workflow

A visual representation of the process is key for success, especially with more complex processes. To create the visual representation you need to determine the workflow of the tasks to complete the project.



For software development a simple example workflow would be…
Analyse -> Design -> Develop -> Test -> Release

Rule #2: Limit Work in Process (WIP)

The Work-In-Process (WIP) is the limit of tasks for each column. The concept is that only a number of things can be worked on at the same time to be done well. There is always an optimal amount of work that can be processed regardless of team size, organization etc. The lower the WIP the quicker bottlenecks or pain points in the process will be revealed but if too low the team will ignore them and learn nothing. Moderate WIP limits is a good compromise with a resilient team to the new process.

Rule #3: Measure and Improve

Similar to other Agile Methodologies improving the process is a constant process and based on metrics. The key metric in Kanban is the WIP and the manager should be focused on looking for the optimal WIP to get the team to reach their maximum potential. Another metric that is brought to light from Kanban boards is the cycle time to complete a task and also the manager should be looking to reduce.

Further Reading:

- https://www.agilealliance.org/glossary/kanban/
- https://resources.collab.net/agile-101/what-is-kanban

Agile Project Planning

Epics

Epics are a group of user stories grouped together to achieve the same business goal/product.

User Stories

A user story is a tool used in Agile development to describe software requirements. User stories are initially written out by the business and then elaborated on by developers. User stories are written out from an end-user perspective. The format for writing out a user story is as follows which would be written on an index card:

As a *<type of user>*, I want *<requirement/feature>*, so that *<purpose/reason>*

Next, after you have a bunch of index cards with user stories on them, you might presume that next is add all the details but you would be mistaken. Each story has 3 elements which are called the 3 C's – the card, the conversation, and the confirmation.

The Card: The card is simply the index card the story is written on, it represents the requirements for planning purposes.

The Conversation: The details of the story are communicated via communication – a verbal exchange of ideas, opinions, and information between the customer and development team.

The Confirmation: This is the customer confirming that the story has been implemented correctly.

When creating a user story it should follow a set of characteristics which can be easily remembered using the INVEST mnemonic…

I: Independent	N: Negotiable	V: Valuable
E: Estimable	S: Specific	T: Timely

Definition of Done

The definition of done is simply the criteria required to complete a story, feature, sprint or release. There will be different DoD at every level of an agile project. The DoD has the following characteristics:

- Check-list of valuable activities: Simple list of required activities that add value to the product which can be checked
- Primary reporting mechanism for team members: A team member can confidently say a story is complete if it meets its DoD
- Not a static definition: DoD changes over time as team and organisation changes

Story Mapping

Story mapping is used to categorize stories into functionality. This makes it easier to identify missing stories and prioritize stories. Grouping stories by feature is also called Epics but story maps are different as they help schedule stories into iterations. Once you have the sprints planned it makes it very easy for the business to see the plan and understand what is going to be done when. Obviously, with Agile this should be kept high level as nothing should be set in stone.

Product Backlog

The Product Backlog is an ordered list of all user stories created by the business. It should consist of every requirement, changes, improvements; it is never complete. As the Product evolves the stories will change but will be can be continuously added to the Product Backlog.

Sprint Backlog

The Sprint Backlog consists of the user stories selected from the Product Backlog to be in the next Sprint. The Sprint backlog is a forecast by the development team what can be expected to be completed in the next Sprint. Only the development team can change the Sprint backlog during a Sprint

Backlog Grooming/Refinement

This is the process of making sure the backlog is prioritised. It is the Product Owners responsibility to ensure the backlog is refined correctly. The backlog stories are prioritised from the top down, so the team will know to choose the story on top of the backlog to work on next. Backlog grooming should be done before estimation sessions so the correct stories are estimated and selected for the next iteration.

Estimation Sessions

Story Points

Story Points assigns a numeric value to a story. Story points should be assign in a sequence such as the Fibonacci 0,1,2,3,5,8,13,21,34 etc to avoid associating stories with hours.

Planning Poker

Planning Poker is a technique used to estimate stories where each person is given deck cards similar to the adjacent picture where each card has a story point value on it. The story is presented by the product owner and the team get to ask any questions. Then simultaneously everyone shows a card representing what they think the estimate should be. If there are differences they are discussed and then another round of estimating is attempted. After the second round if there is not a unanimous decision the highest value is given to the story.

T-Shirt Sizing

T-shirt sizing is a relative sizing technique where a story is assigned a value of extra-small, small, medium, large and extra-large

Iteration & Release Planning

Release planning is done before any iteration planning. The release is planned for a meeting where all stakeholders are represented to ensure their priorities are clearly expressed. The goal is to determine which stories will be delivered in the next release. During release planning the following will occur:

- Assess the prioritised backlog and the estimates on stories.
- Sort the stories by release.
- Refine the roadmap for the upcoming release.

At the end of the meeting, there will be a clear understanding of the release goal and what stories are delivered within the release. The iteration planning then is done by the development team, the product owner and other stakeholders/subject matter experts as needed. Compared to the release planning iteration planning is a lot more detail and estimates will be a lot more accurate. The customer's role will be limited to answering questions for the development team. The Product Owner will start the meeting by describing what they want in the iteration. The development team then commit to what they think is achievable in the iteration. Always remember:

- The Product Owner has the final say on the priorities for the iteration
- The Development Team has the final say on the amount of work to be done in an iteration.

The iteration planning process will follow this process:

1. Discuss user stories in backlog
2. Select user stories for backlog
3. Define acceptance criteria and write acceptance tests
4. Break down user stories into tasks
5. Estimate the tasks.

Agile Project Execution

Daily Stand-ups

The daily stand-up is a time-boxed 15-minute meeting held by the team Agile team every morning. Each team member will keep their update at a high level and should answer the following three questions:

1. What they did since the last time they talked.
2. What they plan on doing before the next meeting.
3. Are there are impediments or blockers preventing them from moving forward.

Only team members will talk not the product owner or any other stakeholders. The product owner can answer the team's questions and the Scrum Master/Team Leader will be responsible for removing impediments.

To further understand why only the team members only talk check out the below cartoon about the pig and chicken. In simple terms, if the pig and chicken open breakfast restaurant together selling bacon and eggs, pig (team) will have it all to lose and the chicken (business) will only be involved.

Timeboxing

Examples of a timeboxing activities that are restricted to certain time period as investigations or documentation. Examples would be an architectural spike is when research/investigation needs to be conducted on an area of the system, technology or application. Another example would be risk-based spike is used to allow the team to eliminate or minimise a risk.

Continuous Integration

Continuous Integration is a practice used by software developers to frequently incorporate new code into their projects code repository. It is all about maintaining quality all the time, throughout the project regardless having multiple developers working on different parts of the code at the same time.

It involves automatically integrating and running a regression test every time somebody does a check in. This is likely to happen several times a day. Running an automated regression test frequently means defects are highlighted soon after they are introduced (i.e. when the build goes Red, i.e. fails). The team's top priority is to get the build Green again.

Reviews

Reviews are usually done by experts from outside the team. Their opinions are usually important and should be followed, or at least considered, by the team. At a certain level, reviews from outside the Agile team can impact the level of "pure" self-organization, since review brings in more people who can evaluate the product increment from different viewpoints. But will a review really break the agility and self-organization of the team? I believe not. Reviews are needed for different reasons. If they're not needed or aren't beneficial to the organisation, of course, they should be reduced or removed from the life cycle.

Retrospectives

After each release/iteration, the Scrum Master/Team Leader should hold a retrospective which can also be called an introspective. The aim of the retrospective is to improve methods and teamwork by reviewing the past iteration/release. There are 3 core questions that should be answered in every retrospective:

1. What is going well
2. What areas could be improved
3. What should we be doing differently

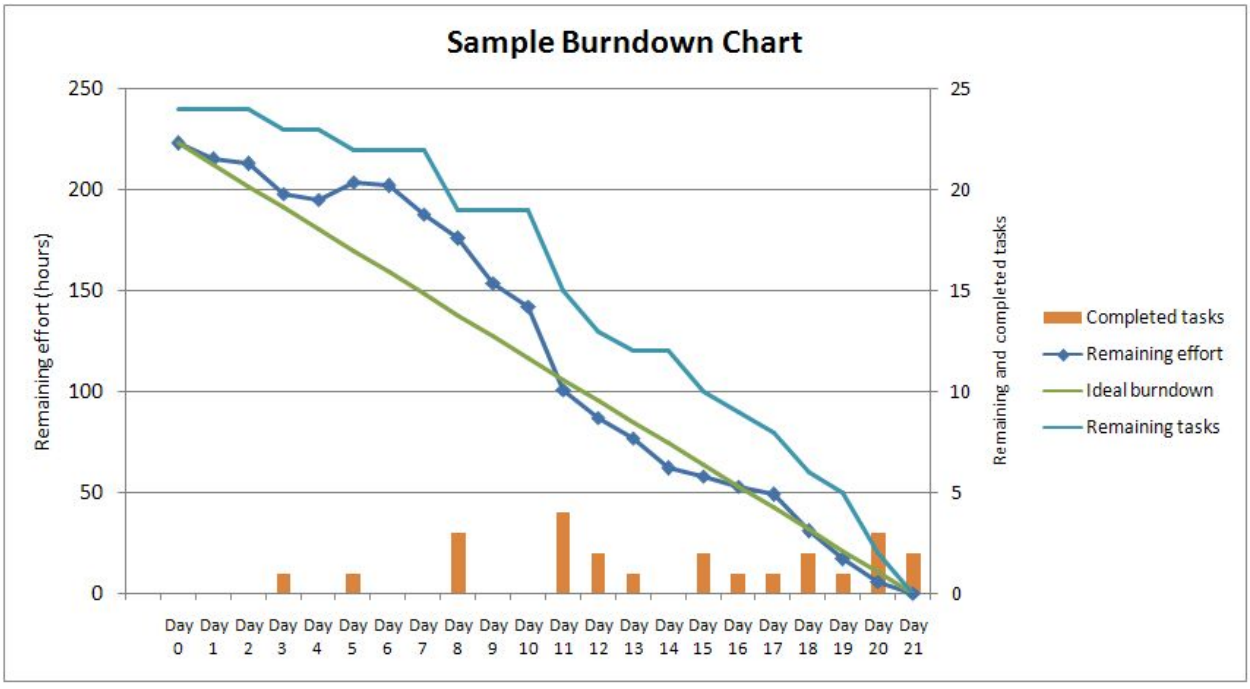
Generally, a retrospective is time-boxed to 2 hours and is based on brainstorming solutions and commit to a solution and discuss again, if success adds to the process.

Agile Project Monitoring

Work in Progress Limits

Work In Progress limits is used to limit the amount of stories allowed in each column on a Kanban board. WIP limits are a strategy for detecting preventing bottlenecks. The WIP limits are agreed by the team before a project starts and enforced by the team facilitator. When a WIP limit has been hit for a certain task the team stops and works together to clear the bottleneck. John Little created Little's Law a mathematical formula based on queueing theory that can be used to analyse work queues on Cumulative Flow Diagram. The formula proves that the duration of work queue is dependent on its size which is why WIP Limits are so important.

Burn Down Charts



The burn down chart is a quick way of determining if sprint is going to plan or not. The diagonal line running from top to bottom represents ideal burn down and is compared to the jotted lines of actual work done. The burn down chart is widely used by teams following the Scrum development methodology.

Resources

THE AGILE SAMURAI
pragprog.com/book/jtrap/the-agile-samurai
One of my favorite books on Agile: it provides not only a great overview, but also an in-depth look at how to run agile properly and successfully.

THE AGILE MANIFESTO
agilemanifesto.org
The founding fathers of Agile.

AGILE & LEAN SOFTWARE DEVELOPMENT LINKED GROUP
linkedin.com/groups/Agile-Lean-Software-Development-37631
Good discussions and Q&A

THE LEAN STARTUP
theleanstartup.com
Book and website dedicated to Lean thinking. Agile and the Lean methodology complement each other perfectly.

PIVOTAL TRACKER
pivotaltracker.com
The popular tracking tool that helps you manage your Agile projects. The feature set is robust, but the User Interface could be improved to make this a go-to tool.

TRELLO
trello.com
A simple online tool to help you manage the project backlog. Trill is like a set of digital post-it notes that you can easily rearrange.

ASANA
asana.com
Another popular management tool. Asana offers a more thorough feature set than Trello, but the interface is not as intuitive

BROUGHT TO YOU BY:

CONTINUAL.DELIVERY

www.continual.delivery

0490-932-552

shane@continual.delivery