



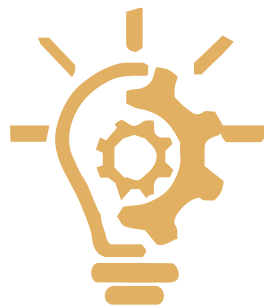
LEAN
SOLUTIONS
ACADEMY

LEAN SIX SIGMA TRAINING

SCRUM PARTICIPANT GUIDE

LEARNING OBJECTIVES

1. Learn the origins of Agile and Scrum
2. Understand the roles on a Scrum Team, Scrum Framework and best practices
3. Understand how Agile and Scrum can be applied across disciplines and industries
4. Apply Agile and Scrum principles in real world scenarios to enhance learning



Reflection:

What are you hoping to get from this course?

PARTNERSHIP AGREEMENTS

These are a few of the agreements that will help us create an inclusive learning environment which we will revise (if needed) as a team. This is a “badge free” and safe space. Meaning, we are all here to learn. Our varied experiences enhance the learning however, titles and organizational hierarchy do not dictate the discussion path.



PARTICIPATE

Actively participate and remain engaged in the learning process for the duration of the course.



BE RESPECTFUL

Show respect for the various points of view presented, regardless of whether you agree.



ASK

If you have a question - ask!
You're probably not the only person who's curious.



HAVE FUN!

You will get out of the session what you put in. Bring your curiosity, positive energy and real-world examples.

INTRODUCTION TO AGILE

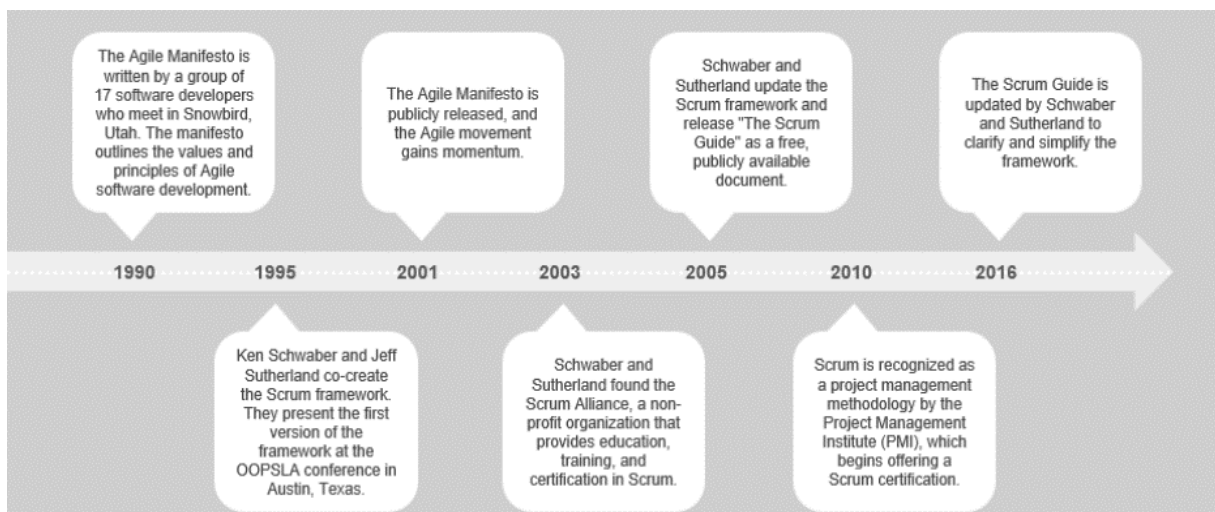
Agile is a software development philosophy and approach that emphasizes flexibility, collaboration, and continuous improvement. It originated as a response to the limitations and drawbacks of traditional, plan-driven project management methodologies.

Traditional design and development steps are separated into distinct phases. Work products are produced at each phase and handed off to the next

Risks:

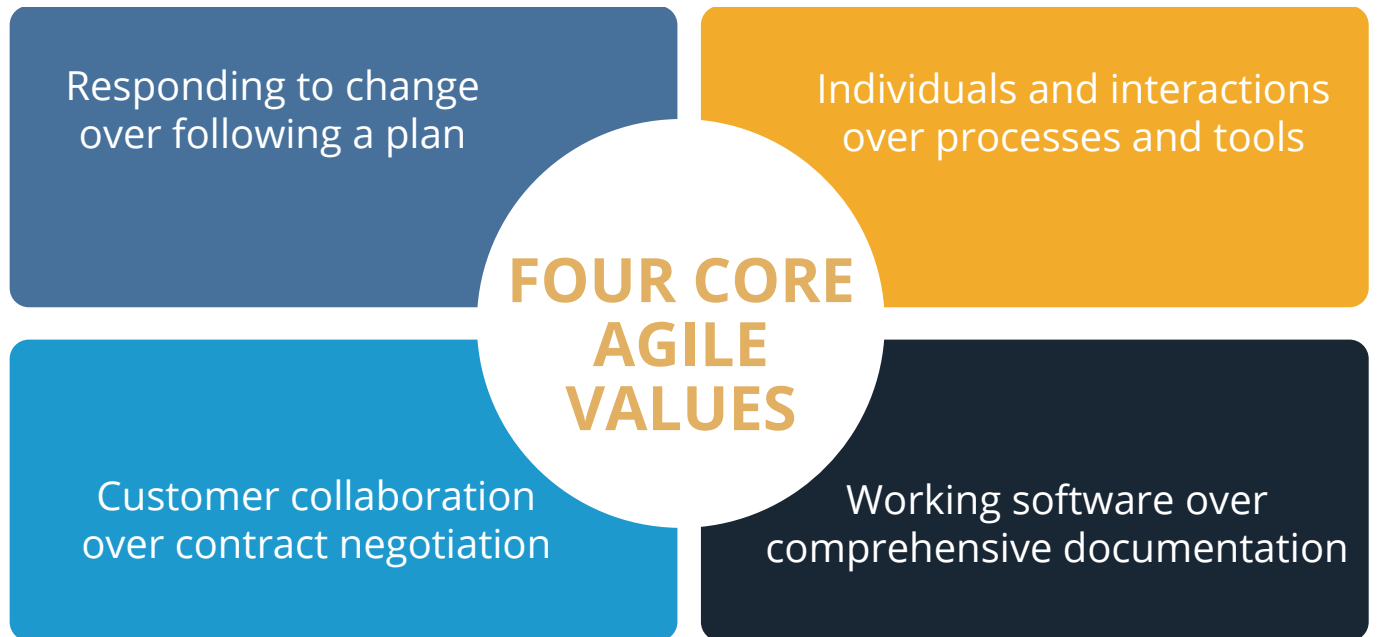
- Errors in each phase are passed to the next
- Time overruns usually come out of final phases – development and test
- Poor quality is compounded by upstream problems in requirements and design
- Accumulates technical debt

The Agile philosophy and principles are described in the Agile Manifesto, which was developed by a group of software developers in 2001.



These principles have become so widely utilized, that they've surpassed strict use in the technology industry and are now used across disciplines and industries.

THE AGILE MANIFESTO



Individuals & Interactions	<i>Over</i>	Processes and Tools
Working Software	<i>Over</i>	Comprehensive Documentation
Customer Collaboration	<i>Over</i>	Contract Negotiation
Responding to Change	<i>Over</i>	Following a Plan

12 AGILE MANIFESTO PRINCIPLES

1. Customer satisfaction through early and continuous delivery of valuable software
2. Welcome changing requirements, even late in development
3. Deliver working software frequently, with a preference for shorter timescales
4. Collaborate with customers and stakeholders throughout the project
5. Build projects around motivated individuals and give them the resources and support they need
6. Use face-to-face communication as much as possible
7. Measure progress primarily through working software
8. Maintain a sustainable pace of work and prioritize work-life balance
9. Keep things simple and minimize unnecessary work
10. Allow self-organizing teams to make decisions
11. Reflect on team performance and continuously improve
12. Focus on technical excellence and good design

Agile is based on Lean Principles

Customer Value

Focus on delivering value to the customer, and continuously identify and meet their changing needs

Continuous Improvement

Continuously identify and eliminate waste and inefficiencies in the development process, and strive for ongoing improvement in processes, products, and services

Respect for People

Encourage collaboration, teamwork, and respect for the people who are involved in the development process

Flow

Streamline the flow of work and minimize delays, bottlenecks, and handoffs

Pull

Work is pulled through the development process in response to demand, rather than pushed through based on predetermined schedules



INTRODUCTION TO SCRUM

Scrum is an Agile framework for software development that is designed to help teams work together more efficiently and effectively. It is an iterative and incremental approach that prioritizes collaboration, flexibility, and continuous improvement.

Scrum Theory

The Scrum theory is based on the principles of Agile software development and emphasizes the iterative and incremental delivery of products

The Scrum framework provides a set of roles, events, and artifacts that enable teams to collaborate and deliver value in an iterative and incremental way

The theory is based on the idea that product development is a complex process that cannot be fully planned and predicted, and that teams must be able to adapt to changing requirements and circumstances. The theory emphasizes the importance of transparency, inspection, and adaptation in order to improve the product and the process over time



TRANSPARENCY

All aspects of the process and the product are visible to everyone involved.



INSPECTION

The team regularly reviews and evaluates the product and the process to identify areas for improvement.



ADAPTATION

The team makes changes based on the results of the inspection to continually improve the product and the process.

Scrum is based on the theory of empirical process control, which relies on transparency, inspection, and adaptation.

5 CORE VALUES OF SCRUM

COMMITMENT

Scrum teams are committed to achieving their goals and delivering value to their customers. Team members are expected to be reliable and to work together to ensure that commitments are met.

FOCUS

Scrum teams focus on their goals and work collaboratively to deliver value. They prioritize their work and stay focused on what is most important, in order to deliver the highest value to their customers.

OPENESS

Scrum teams are transparent and open in their communication and work processes. They share information freely, welcome feedback, and work together to solve problems.

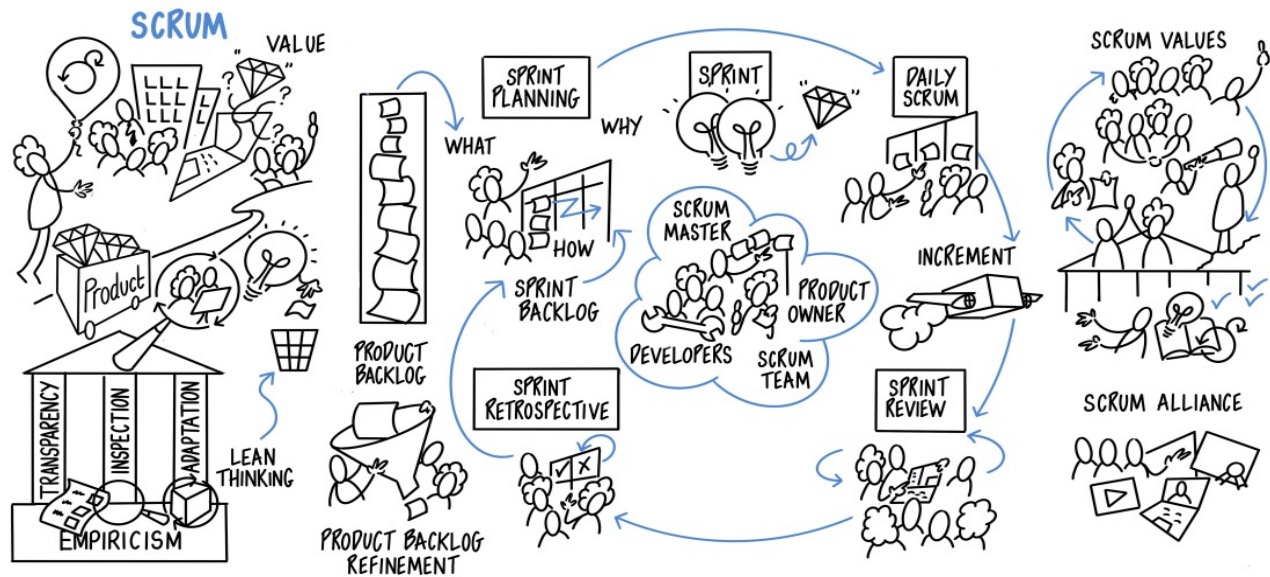
RESPECT

Scrum teams show respect for each other and for their customers and stakeholders. They value diversity and work together to create an inclusive and supportive environment.

COURAGE

Scrum teams have the courage to take on complex challenges and to make difficult decisions. They are willing to speak up and take risks in order to achieve their goals and deliver value to their customers.

SCRUM OVERVIEW



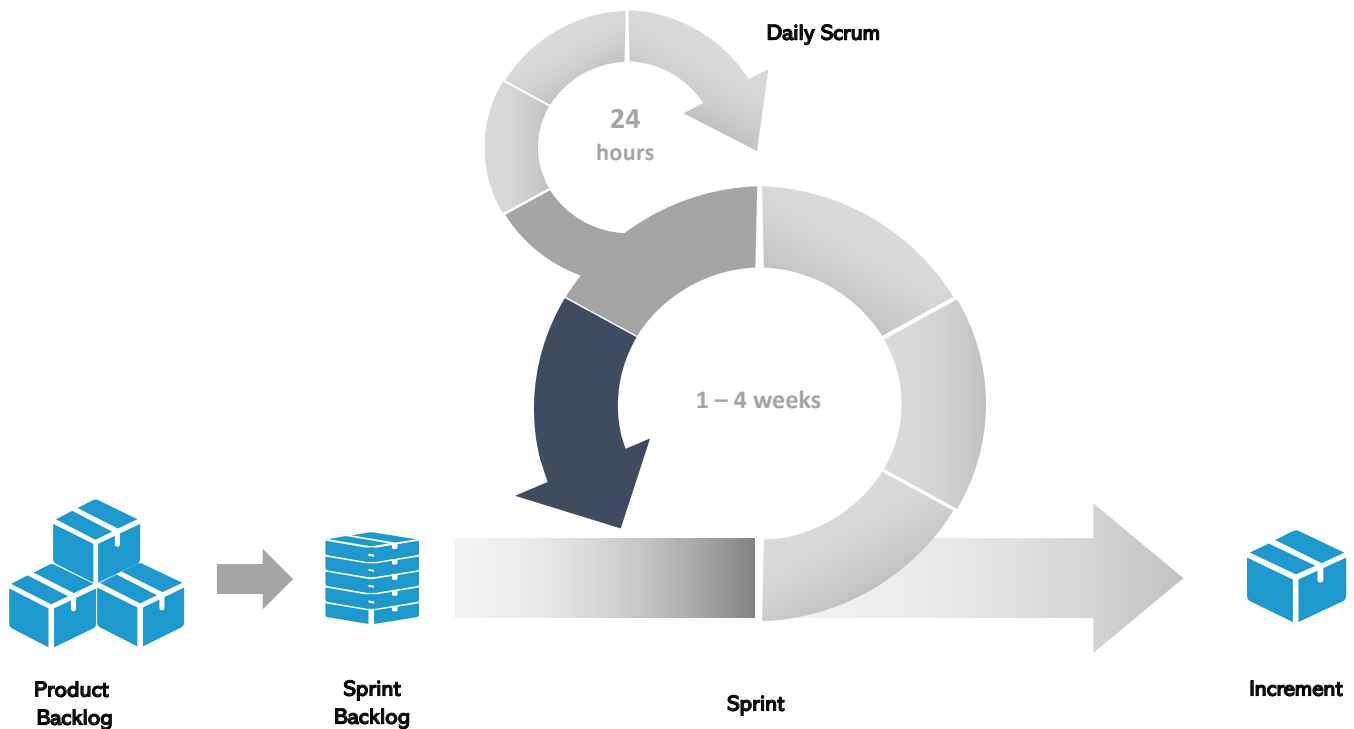
Scrum Is...

- Scrum is a specific Agile framework that includes a set of roles, events, artifacts, and rules that help teams manage their work
- Scrum emphasizes iterative development, regular feedback, and continuous improvement
- Scrum includes key roles such as the Scrum Master, Product Owner, and Development Team, as well as events like the Sprint, Daily Scrum, and Sprint Review

Scrum Isn't...

- Scrum is not a silver bullet that guarantees project success
- Scrum is not a one-size-fits-all solution that will work for every project or team
- Scrum is not a replacement for good project management practices or a way to avoid planning and documentation
- Scrum is not the only Agile framework, but it is one of the most popular and widely used

CHARACTERISTICS OF SCRUM

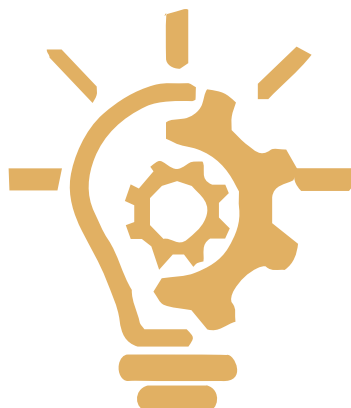


- Self-organizing teams
- Product progresses in a series of “sprints” (1-4 weeks in duration)
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”
- Work in Progress (WIP) for the sprint is estimated by the Development Team



KNOWLEDGE CHECK

1. The three pillars of empiricism are _____, _____ and _____.
2. Scrum is an intentionally lightweight framework / methodology. (Circle one)
3. Scrum is based on _____ principles.
4. Scrum is both _____ and _____.
5. The final delivery within the scrum framework is called an _____.



BENEFITS & DRAWBACKS OF AGILE & SCRUM

Benefits

Flexibility & Adaptability

Agile and Scrum are designed to accommodate changes in requirements and priorities, which can help teams respond more effectively to new information or unexpected issues.

Collaboration & Teamwork

The emphasis on communication, feedback, and collaboration can help teams work together more effectively and achieve better results.

Customer Satisfaction

By involving customers and stakeholders in the development process, teams can ensure that the product meets their needs and expectations.

Early & Frequent Delivery

By delivering working software incrementally and frequently, teams can get feedback early and make adjustments as needed.

Continuous improvement

Agile and Scrum provide opportunities for teams to reflect on their performance and identify areas for improvement, which can lead to better results over time.

Drawbacks

Learning curve

Agile and Scrum require a significant shift in mindset and practices for teams used to traditional, plan-driven approaches, which can take time to adapt to.

Lack of predictability

The focus on flexibility and adaptability can make it difficult to predict the outcome or timeline of a project, which can be challenging for stakeholders who need to plan and budget accordingly.

Overemphasis on process

Teams can sometimes become too focused on following the process rather than delivering value, which can lead to inefficiencies or delays.

Resource requirements

Agile and Scrum require significant time and effort from team members, which can be challenging for teams with limited resources or competing priorities.

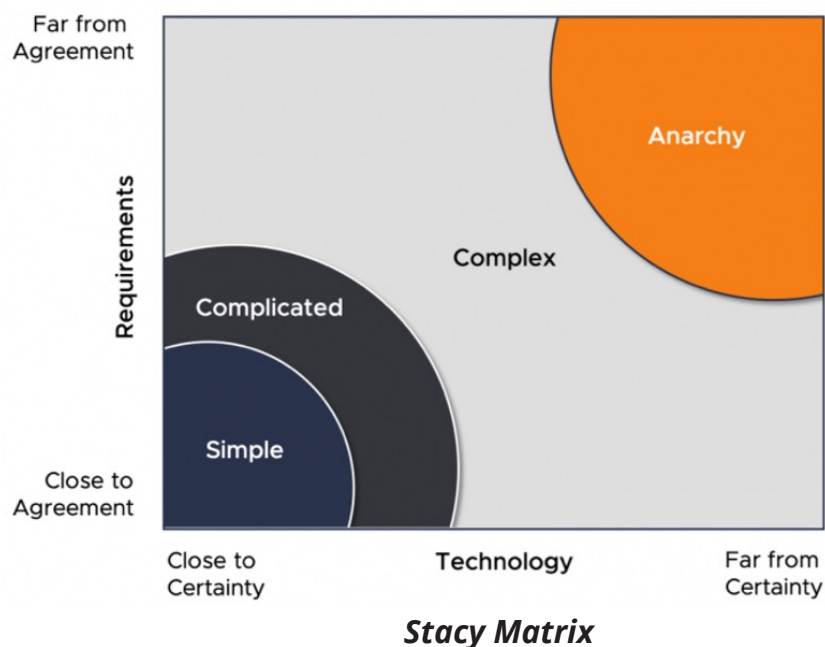
Ineffective implementation

Agile and Scrum can be implemented poorly or inconsistently, which can lead to confusion, miscommunication, or suboptimal results.

PROJECT TAILORING

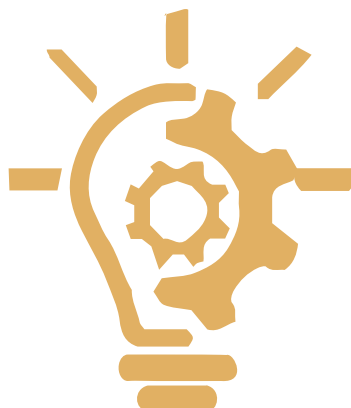
A tailored project management methodology produces direct and indirect benefits to organizations—generally, the greater the level of project management methodology tailoring, the greater the level of project success. Additional benefits of a tailored approach to a project management methodology include:

- More commitment from team members who helped to create it.
- A customer-oriented focus as the needs of the customer are an important influencing factor in its development.
- A more efficient use of project resources.
- In the absence of an appropriately tailored project management methodology, individuals will tailor their own solutions. Here are some signs that the methodology is not tailored correctly:
- Project team members are not using the methodology and are managing projects without any structure or uniform process at all.
- Project team members are independently modifying the methodology without guidance and are developing their own unique versions, eliminating the efficiencies gained by standardization.
- The methodology features processes for the sake of process with little thought given to whether or not they deliver benefits and contribute to project success.
- It is a one-size-fits-all approach to projects of differing types, sizes and complexity.



KNOWLEDGE CHECK

1. Determining which project execution approach to take based on environmental variables is called _____.
2. One of the primary disadvantages of scrum is that it's less predictable in terms of _____ and _____.
3. One of the primary advantages of scrum is that it allows for _____ and _____ delivery.
4. There are _____ agile manifesto principles.
5. Scrum has __ Core Values



353 SCRUM FRAMEWORK

Without embracing the Scrum Values and operating according to the 3-5-3 framework, you really aren't doing scrum.

3 Roles

Scrum Master:

The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization. They are accountable for the Scrum Team's effectiveness by enabling the Scrum Team to improve its practices within the Scrum framework. Scrum Masters are true leaders who serve the Scrum Team and the broader organization.

Development Team:

Also known simply as the Team, it is comprised of people who work on Sprint Backlog Items. It acts as 'one team' and has all the skills needed to produce a working tested increment each Sprint. The development team is:

- Self-Organizing
- Cross-Functional
- Accountable
- Small with 3 - 9 team members

Product Owner:

One of three Scrum Roles. Responsible for creating a compelling product vision that is executable. Other responsibilities include curating and prioritizing a Product Backlog, spending 50% of their time with customers and stakeholders, and 50% working closely with the team

5 Events

Sprint:

One of the five Scrum Events. It is a short, consistent cycle no longer than four weeks. The goal is to have an iteration short enough to keep the team focused but long enough to deliver a meaningful increment of work. All other Scrum Events take place during a Sprint. Once a Sprint is finished, the next begins.

Sprint Planning:

One of the five Scrum Events. Where the work to be performed in the Sprint is planned. This event is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that



attendants understand its purpose.

Daily Scrum:

One of the 5 Scrum Events, this meeting lasts no more than fifteen minutes and happens every workday at the same appointed time and place. Everyone on the team attends. At this meeting, the information needed to assess progress is presented, and any impediments are noted. This information may result in re-planning of the Sprint.

Sprint Review:

One of the five Scrum Events. Held at the end of the Sprint to allow customers and stakeholders to inspect the Increment, give feedback, and for the Scrum Team to adapt the Product Backlog if needed. This is at most a four-hour meeting for one-month Sprints.

Sprint Retrospective:

One of the five Scrum Events. An opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint. This event occurs after the Sprint Review and prior to the next Sprint Planning. At most a three-hour meeting for one-month Sprints.

3 Artifacts

Product Backlog:

A priority ordered list of everything that is known to be needed in the product. It is the source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

Sprint Backlog:

The set of Product Backlog Items selected for the Sprint, plus a plan for delivering the Increment and realizing the Sprint Goal. It includes at least one Improvement identified at the last Sprint Retrospective. It makes visible all of the work the Team identifies as necessary to meet the Sprint Goal.

Product Increment:

An Increment (sometimes referred to as a 'Potentially Shippable Product') is the value delivered for the customer via the Product Backlog Items completed during a Sprint. Each Increment should interface seamlessly with all prior Increments and stand alone as a distinct addition of value to the Product. While more than one Increment may be created in a Sprint, all work must meet the Definition of Done to be considered complete.

Other - Backlog Refinement:

The ongoing process of adding detail, estimates, and order to items in the Backlog. Refinement usually consumes no more than 10% of the capacity of the Team. However, Backlog Items can be updated at any time by the Product Owner or at the Product Owner's discretion.

KNOWLEDGE CHECK

1. The 353 Scrum Framework refers to 3_____, 5_____ and 3_____.
2. The _____ _____ is responsible for setting the product vision and being connected to voice of customer.
3. The _____ _____ is the servant leader and coach to the team.
4. The _____ _____ is responsible for how work is delivered.
5. Scrum artifacts all contain _____ that must be made and kept throughout the sprint.
6. Teams should be _____ _____ and complete work based on their expertise.



KNOWLEDGE CHECK

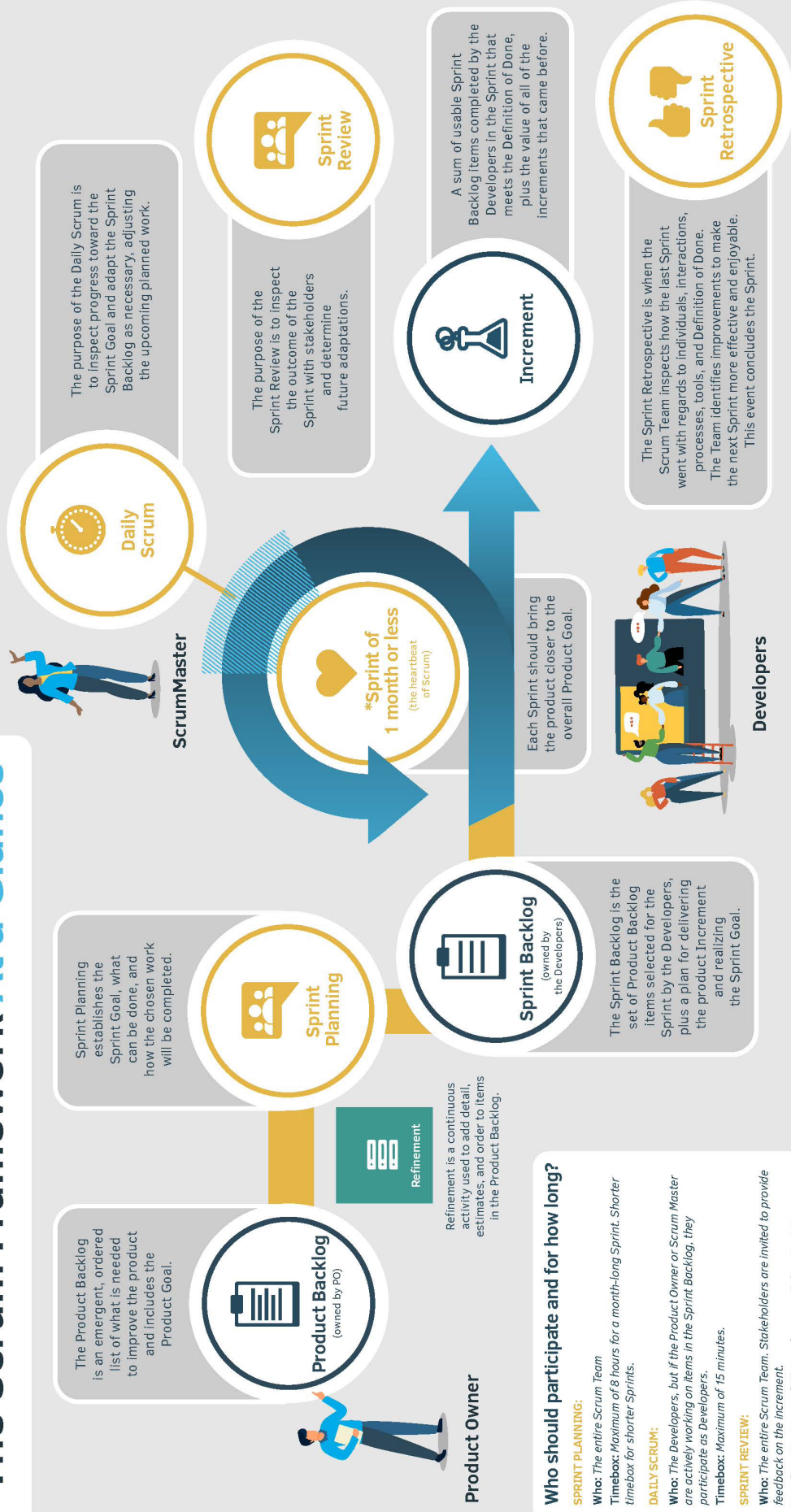
1. _____ is an umbrella term, used to describe various methodologies / frameworks that help organizations deliver value to customers most efficiently.
2. Maintain a sustainable pace of work and prioritize work-life.
3. balance is a principle of the _____.
4. The five core values of Scrum are _____, _____, _____ and _____.
5. What is the 'Manager' role responsible for in scrum?

6. The Stacy Matrix helps us determine an execution approach based on _____ and _____.



The Scrum Framework At a Glance

Version 5.0



Who should participate and for how long?

SPRINT PLANNING:

Who: The entire Scrum Team

Timebox: Maximum of 8 hours for a month-long Sprint. Shorter timebox for shorter Sprints.

DAILY SCRUM:

Who: The Developers, but if the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

Timebox: Maximum of 15 minutes.

SPRINT REVIEW:

Who: The entire Scrum Team. Stakeholders are invited to provide feedback on the increment.

Timebox: Maximum of 4 hours for a month-long Sprint.

Shorter timebox for shorter Sprints.

SPRINT RETROSPECTIVE:

Who: The entire Scrum Team.

Timebox: Maximum of 3 hours meeting for a month-long Sprint. Shorter timebox for shorter Sprints.



Copyright © 2020 SCRUM ALLIANCE®, Inc. All Rights Reserved.

- = Scrum artifacts that help manage the work
- = Events or ceremonies that occur inside each Sprint
- = Ongoing activity

MAKING THE SCRUM SYSTEM WORK

The Sprint

Sprint is the heart of Scrum. It's a time-box of up to one month (usually 2 weeks) and a container of all other events (Sprint Planning, Daily Scrum, Sprint Review and Sprint Retrospective).

A new Sprint starts immediately after the end of the existing Sprint. During Sprint, the team works together to create a potentially shippable product increment.

- Sprint enables predictability.
- Sprint limits risk.
- Sprint duration should be consistent during the development effort.
- Sprint ends when time box expires.
- During the sprint no changes made in sprint goal, quality goal does not decrease, and scope may be renegotiated between the Product Owner and Development Team as more is learned.

Cancelling a Sprint: A Sprint can be canceled before the Sprint duration is over. Only the Product Owner can cancel the Sprint they have the authority to cancel. The Product owner may do so under influence from the stakeholders, the Development Team, or the Scrum Master. A Sprint would be canceled if the Sprint Goal becomes obsolete.

Sprint Planning

Sprint Planning is the event in which the Product Owner presents the ordered product backlog to the development team. As the word suggests, 'Sprint Planning' means we are going to plan the work to be done in the Sprint.

There are two main parts - 'What' and 'How'.

- 'What' can be done in this Sprint?
- 'How' will the selected work get done?

Summary

- Who will participate - Development Team, Product Owner, and Scrum Master.
- When - At the beginning of the sprint
- Time-box - Maximum of eight hours for the four weeks Sprint.
- Input - Product backlog, latest product increment, the definition of done, team capacity, past performance.
- Output - Sprint goal, sprint backlog, a clear understanding of work to be done during the sprint, transparency.

MAKING THE SCRUM SYSTEM WORK

Daily Scrum

Daily Scrum is a daily standup meeting between Scrum Team. It's a 15-minute time-boxed daily event generally held at the same place and the same time.

In this meeting, development team explains

- What did I do yesterday?
- What will I do today?
- Are there any issues or Impediments?

Development team inspects the progress towards the Sprint goal in the daily standup. The everyday 15-minute meeting with the team helps each member to work together in a self-organized way, and they learn collaboration.

Scrum master makes sure that each member participates in the daily standup. This helps the team in terms of better communication, decision-making, and improved level of knowledge. The team learns inspection and adoption.

Summary

- Who will participate - Development Team, Scrum Master (Product Owner – Optional)
- When - Everyday same place, same time
- Time-box - 15 mins Max.
- Input - 3 questions - "What did I do yesterday?", "What will I do today?" and "Are there any issues or Impediments?"
- Output - Clear idea about the progress towards Sprint goal, any issues or Impediment idea in the earlier stage.

Sprint Review

Sprint Review is held at the end of each Sprint. The main purpose of the Sprint review is to inspect the incremented product created in the Sprint and adapt the product backlog if needed.

The Development Team demonstrates the work that it was done. During the review meeting, Scrum Team and Stakeholders collaborate to discuss what was done in the Sprint and what to do next.

It will give valuable feedback to the subsequent Sprint planning. Sprint review meeting is an informal meeting, not a status update meeting.

Summary

- Who will participate - Development Team, Scrum Master, Product Owner and Stakeholders

- When - End of the Sprint and before sprint retrospective.
- Time-box - Four hours for one month's Sprint.
- Input - Product Increment, Changes to the product backlog during the Sprint.
- Output - Updated product backlog, New Idea, clear picture of tasks and product.

Sprint Retrospective

Sprint Retrospective is the chance for the Scrum team to inspect itself and create a plan for improvements to be taken care in the next Sprint.

The main purpose of the Sprint retrospective is,

- Inspect how the sprint went with regard to process, tool, and people.
- Identify items that went well and potential improvements.
- Create an action plan to implement improvements in the scrum team.
- Scrum Master encourages the team to improve. In the Scrum process framework, practices of processes make tasks more effective and enjoyable for the next Sprint.

Implementing these improvements in the next Sprint is the adaptation to the inspection of the Scrum Team itself.

Summary

- Who will participate - Development Team, Scrum Master, Product Owner
- When - at the end of the sprint after sprint review.
- Time-box - Three hours for one month's Sprint.
- Input - Results from the sprint, Sprint Events
- Output - Lesson learned, Improvements, and action list for the next sprint.

Scrum Artifacts

Product Backlog

Commitment: Product Goal

A priority ordered list of everything that is known to be needed in the product. It is the source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering.

Sprint Backlog

Commitment: Sprint Goal

The set of Product Backlog Items selected for the Sprint, plus a plan for delivering the Increment and realizing the Sprint Goal. It includes at least one Improvement identified at the last Sprint Retrospective. It makes visible all of the work the Team identifies as necessary to meet the Sprint Goal.

Increment

Commitment: Definition of Done

An Increment (sometimes referred to as a 'Potentially Shippable Product') is the value delivered for the customer via the Product Backlog Items completed during a Sprint. Each Increment should interface seamlessly with all prior Increments and stand alone as a distinct addition of value to the Product. While more than one Increment may be created in a Sprint, all work must meet the Definition of Done to be considered complete.



Definition of Done

Team: _____

Product: _____

As a team, before saying that an item of the Sprint Backlog is Done, we agree that it will meet the following:

- ☐ The code is complete and according to developers' standards.
- ☐ The code is refactored.
- ☐ Technical Debt is removed.
- ☐ Meet the acceptance criteria.
- ☐ Code checked-in to the repository.
- ☐ Unit tests are written and green.
- ☐ Test coverage: ____ %.
- ☐ Pair programming.
- ☐ Peer review.
- ☐ Code merge and tagged.
- ☐ Deployed to the development environment.
- ☐ Integration tests are written and green.
- ☐ Deployed to the QA environment.
- ☐ Bugs, improvements, and changes are done.
- ☐ Changes were communicated and updated in the ticket description.
- ☐ Regression tests.
- ☐ Performance tests.
- ☐ Approved by QA.
- ☐ Deployed to UAT.
- ☐ Reviewed and accepted by the Product Owner.
- ☐ Reviewed and accepted by the customer or user(s).
- ☐ Final approval in the Sprint Review or demo by the Product Owner and customer.
- ☐ Deployed to production.
- ☐ Production tests are successful.
- ☐ _____
- ☐ _____
- ☐ _____

Backlog Refinement

Focus on the Goal: If the items in your product backlog don't support your product goal, you might build beautifully executed work that doesn't provide value.

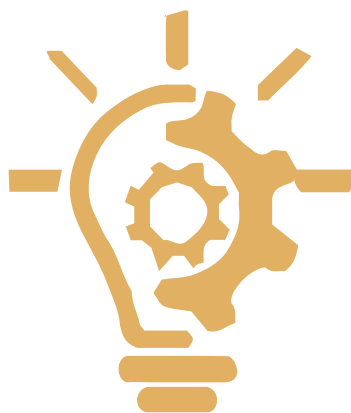
Ask the Right Questions: The exact questions you will want to ask will vary a lot depending on the work domain. These five questions are good starting points for any PBI (product backlog item) being refined.

- What is the desired outcome? What will you be able to do when this PBI is done?
- Who is this for? Who is going to use the value created by this PBI? Be specific.
- What value will this create? What kind of value will we get from this PBI?
- How much effort will it take? To paraphrase Einstein, "if you can't size it, you do not truly understand it."
- "What problem are you trying to solve?"



KNOWLEDGE CHECK

1. The sprint is over when the _____ expires.
2. Agile is an _____ approach to development.
3. A term used for combining Scrum and Kanban approaches in delivery is _____-_____.
4. High Functioning and _____ _____ teams are critical to scrum success.
5. The appropriate time box for an iteration is 1 to 4 _____?
6. The recommended size for a scrum team is ____ or less members.



RESOURCES



REFERENCES

Achieving Business Agility and Agile Planning | Scaled Agile

www.pmi.org/learning/library/tailoring-benefits-project-management-methodology-11133

Manifesto for Agile Software Development (agilemanifesto.org)

OEConsulting.com.sg

Scrum Alliance Certification | Transform your workplace

Scrum Inc. Home - Better Results. Starting Now.

The Scrum Guide the Definitive Guide to Scrum: Rules of the Game

www.controlchaos.com

www.mountangoatsoftware.com/scrum

NOTES



NOTES



NOTES



GLOSSARY

A

Acceptance Test Driven Development (ATDD)

Acceptance Test Driven Development (ATDD) involves team members with different perspectives (customer, development, testing) collaborating to write acceptance tests in advance of implementing the corresponding functionality.

Acceptance Testing

An acceptance test is a formal description of the behavior of a software product, generally expressed as an example or a usage scenario. A number of different notations and approaches have been proposed for such examples or scenarios. In many cases the aim is that it should be possible to automate the execution of such tests by a software tool, either ad-hoc to the development team or off the shelf.

Antipattern

Antipatterns are common solutions to common problems where the solution is ineffective and may result in undesired consequences.

Automated Build

In the context of software development, build refers to the process that converts files and other assets under the developers' responsibility into a software product in its final or consumable form. The build is automated when these steps are repeatable, require no direct human intervention, and can be performed at any time with no information other than what is stored in the source code control repository.

B

Backlog

See product backlog.

Backlog Refinement

Backlog Refinement (formerly Backlog Grooming) is when the product owner and some, or all, of the rest of the team, refine the backlog on a regular basis to ensure the backlog contains the appropriate items, that they are prioritized, and that the items at the top of the backlog are ready for delivery. (read more)

Behavior Driven Development (BDD)

BDD is a practice where members of the team discuss the expected behavior of a system in order to build a shared understanding of expected functionality.

Burndown Chart

Burndown charts and burnup charts track the amount of output (in terms of hours, story points, or backlog items) a team has completed across an iteration or a project.

Business Agility

Business agility is the ability of an organization to sense changes internally or externally and respond accordingly in order to deliver value to its customers.

C

Collective Ownership

Collective code ownership is the explicit convention that every team member can make changes to any code file as necessary: either to complete a development task, to repair a defect, or to improve the code's overall structure.

Continuous Deployment

Continuous deployment aims to reduce the time elapsed between writing a line of code and making that code available to users in production. To achieve continuous deployment, the team relies on infrastructure that automates and instruments the various steps leading up to deployment, so that after each integration successfully meeting these release criteria, the live application is updated with new code.

Continuous Integration

Continuous Integration is the practice of merging code changes into a shared repository several times a day in order to release a product version at any moment. This requires an integration procedure which is reproducible and automated.

CRC Cards

Class Responsibility Collaborator (CRC) Cards are an object oriented design technique teams can use to discuss what a class should know and do and what other classes it interacts with.

Customer Development

Customer development is a four-step framework that provides a way to use a scientific approach to validate assumptions about your product and business. ([learn more](#))

D

Daily Meeting

The daily meeting is one of the most commonly practiced Agile techniques and presents opportunity for a team to get together on a regular basis to coordinate their activities.

Definition of Done

The definition of done is an agreed upon list of the activities deemed necessary to get a product increment, usually represented by a user story, to a done state by the end of a sprint.

Definition of Ready

Definition of Ready involves creating clear criteria that a user story must meet before being accepted into an upcoming iteration. This is typically based on the INVEST matrix.

E

Epic

An epic is a large user story.

Estimation

In software development, an "estimate" is the evaluation of the effort necessary to carry out a given development task; this is most often expressed in terms of duration.

Exploratory Testing

Exploratory testing is, more than strictly speaking a “practice,” a style or approach to testing software which is often contrasted to “scripted testing.”

Extreme Programming

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

F

Facilitation

A facilitator is a person who chooses or is given the explicit role of conducting a meeting.

Frequent Releases

An Agile team frequently releases its product into the hands of end users, listening to feedback, whether critical or appreciative.

G

Given When Then

The Given-When-Then formula is a template intended to guide the writing of acceptance tests for a User Story: (Given) some context, (When) some action is carried out, (Then) a particular set of observable consequences should obtain.

I

Incremental Development

In an Agile context, Incremental Development is when each successive version of a product is usable, and each builds upon the previous version by adding user-visible functionality.

Information Radiators

“Information radiator” is the term for any of a number of visual displays which a team places in a highly visible location, so that all team members can see the latest information at a glance.

Integration

“Integration” (or “integrating”) refers to any efforts still required for a project team to deliver a product suitable for release as a functional whole.

INVEST

The acronym INVEST stands for a set of criteria used to assess the quality of a user story. If the story fails to meet one of these criteria, the team may want to reword it.

Iteration

An iteration is a timebox during which development takes place. The duration may vary from project to project and is usually fixed.

Iterative Development

Agile projects are iterative insofar as they intentionally allow for “repeating” software development

activities, and for potentially “revisiting” the same work products (the phrase “planned rework” is sometimes used; refactoring is a good example).

K

Kanban

The Kanban Method is a means to design, manage and improve flow for knowledge work and allows teams to start where they are to drive evolutionary change.

Kanban Board

A Kanban Board is a visual workflow tool consisting of multiple columns. Each column represents a different stage in the workflow process.

L

Lead Time

Lead Time is the time between a customer order and delivery. In software development, it can also be the time between a requirement made and its fulfillment.

M

Milestone Retrospective

A Milestone Retrospective is a team’s detailed analysis of the project’s significant events after a set period of time or at the project’s end.

Minimum Marketable Feature (MMF)

A Minimum Marketable Feature is a small, self-contained feature that can be developed quickly and that delivers significant value to the user.

Minimum Viable Product (MVP)

A Minimum Viable Product is, as Eric Ries said, the “version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.”

Mob Programming

Mob Programming is a software development approach where the whole team works on the same thing, at the same time, in the same space, and at the same computer.

Mock Objects

Mock Objects (commonly used in the context of crafting automated unit tests) consist of instantiating a test-specific version of a software component.

N

Niko-niko Calendar

A Niko-niko Calendar is updated daily with each team member’s mood for that day. Over time the calendar reveals patterns of change in the moods of the team, or of individual members.

O

Open Space

In Open Space meetings, events, or conferences, participants create and manage their own agenda of parallel sessions around a specific theme.

P

Pair Programming

Pair programming consists of two programmers sharing a single workstation (one screen, keyboard and mouse among the pair).

Personas

Personas are synthetic biographies of fictitious users of the future product.

Planning Poker

An approach to estimation used by Agile teams. Each team member “plays” a card bearing a numerical value corresponding to a point estimation for a user story.

Points (estimates in)

Agile teams generally prefer to express estimates in units other than the time-honored “man-hours.” Possibly the most widespread unit is “story points.”

Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.

Product Owner

The product owner is a role created by the Scrum Framework responsible for making sure the team delivers the desired outcome.

Project Chartering

A high-level summary of the project’s key success factors displayed on one wall of the team room as a flipchart-sized sheet of paper.

Q

Quick Design Session

When “simple design” choices have far-reaching consequences, two or more developers meet for a quick design session at a whiteboard.

R

Refactoring

Refactoring consists of improving the internal structure of an existing program’s source code, while preserving its external behavior.

Relative Estimation

Relative estimation consists of estimating tasks or user stories by comparison or by grouping of items of equivalent difficulty.

Retrospective

The team meets regularly to reflect on the most significant events that occurred since the previous such meeting, and identify opportunities for improvement.

Rule of Simplicity

Rules of Simplicity is a set of criteria, in priority order, proposed by Kent Beck to judge whether some source code is “simple enough.”

S

Scrum

Scrum is a process framework used to manage product development and other knowledge work.

Scrumban

Scrumban is a mixture of Scrum and Kanban. For a more in-depth look at Scrumban, please see this post written by the creator of the Scrumban Method, Corey Ladas.

Scrum Master

The scrum master is responsible for ensuring the team lives agile values and principles and follows the practices that the team agreed they would use.

Scrum of Scrums

A technique to scale Scrum up to large groups (over a dozen people), consisting of dividing the groups into Agile teams of 5-10.

Sign Up for Tasks

Members of an Agile development team normally choose which tasks to work on, rather than being assigned work by a manager.

Simple Design

A team adopting the “simple design” practice bases its software design strategy on a set of “simple design” principles.

Sprint Backlog

A sprint backlog is the subset of product backlog that a team targets to deliver during a sprint in order to accomplish the sprint goal and make progress toward a desired outcome.

Sprint Planning

Sprint planning is an event that occurs at the beginning of a sprint where the team determines the product backlog items they will work on during that sprint.

Story Mapping

Story mapping consists of ordering user stories along two independent dimensions.

Story Splitting

Splitting consists of breaking up one user story into smaller ones, while preserving the property that each user story separately has measurable business value.

Sustainable Pace

The team aims for a work pace that they would be able to sustain indefinitely.

T

Task Board

The most basic form of a task board is divided into three columns labeled “To Do,” “In Progress,” and “Done.” Cards are placed in the columns to reflect the current status of that task.

Test Driven Development (TDD)

“Test-driven development” is a style of programming in which three activities are tightly interwoven: coding, testing (in the form of writing unit tests) and design (in the form of refactoring).

Team

A “team” in the Agile sense is a small group of people, assigned to the same project or effort, nearly all of them on a full-time basis.

Team Room

The team (ideally the whole team, including the product owner or domain expert) has the use of a dedicated space for the duration of the project, set apart from other groups’ activities.

Three C’s

“Card, Conversation, Confirmation” is a formula that captures the components of a User Story.

Three Amigos

Three amigos refers to the primary perspectives to examine an increment of work before, during, and after development. Those perspectives are Business, Development, and Testing.

Three Questions

The daily meeting is structured around some variant of the following three questions: What have you completed? What will you do next? What is getting in your way?

Timebox

A timebox is a previously agreed period of time during which a person or a team works steadily towards completion of some goal.

U

Ubiquitous Language

Striving to use the vocabulary of a given business domain, not only in discussions about the requirements for a software product, but in discussions of design as well and all the way into “the product’s source code itself.”

Unit Testing

A unit test is a short program fragment written and maintained by the developers on the product team, which exercises some narrow part of the product’s source code and checks the results.

Usability Testing

Usability testing is an empirical, exploratory technique to answer questions such as “how would an end user respond to our software under realistic conditions?”

User Stories

In consultation with the customer or product owner, the team divides up the work to be done into

functional increments called “user stories.”

User Story Template

The user story template is one of the most commonly recommended aids to write user stories: As a ... I want ... So that ...

V

Velocity

At the end of each iteration, the team adds up effort estimates associated with user stories that were completed during that iteration. This total is called velocity.

Version Control

Version control is not strictly an Agile “practice” insofar as it is now widespread in the industry as a whole. But it is mentioned here for several reasons.

X

XP – Extreme Programming

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.



LEAN SOLUTIONS ACADEMY

*Email: office@patrickadamsconsulting.com
Go online: findleansolutions.com*