

AGILE DEVELOPMENT notes...

Software Development Life Cycle (SDLC)

A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle which represents all the methods required to make a software product.

Stage 1: Feasibility it is based on many purposes to analyze whether software product will be right in terms of development, implantation, contribution of project to the organization etc.

Types of Feasibility Study: The feasibility study mainly concentrates on below five mentioned areas:

1. Technical Feasibility – In Technical Feasibility current resources both hardware software along with required technology are analyzed to develop project. It gives report whether there exists correct required resources and technologies which will be used for project development.

2. Operational Feasibility – Operational feasibility analyzes degree of providing service to requirements.

3. Economic Feasibility – In Economic Feasibility detailed analysis of cost of project like hardware and software resource required, design and development cost and benefit of the project is analyzed.

4. Legal Feasibility – In Legal Feasibility study project is analyzed in legality point of view which includes analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc.

5. Schedule Feasibility – In Schedule Feasibility Study mainly timelines/deadlines is analyzed for proposed project which includes how many times teams will take to complete final project.

Stage 2: Requirement Analysis. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. Software requirement means requirement that is needed by software to increase quality of software product.

1. **Problem Recognition:** The main aim of requirement analysis is to fully understand main objective of requirement that includes why it is needed, does it add value to product, will it be beneficial, does it increase quality of the project, and does it will have any other effect.

Stage 3: Software Design Process the design phase of software development deals with transforming the customer requirements as described in the SRS documents into a form implementable using a programming language. The software design process can be divided into the following three levels of phases of design:

1. Interface Design

2. Architectural Design

3. Detailed Design

Interface Design: Interface design is the specification of the interaction between a system and its environment at a high level of abstraction with respect to the inner workings of the system.

Architectural Design

Detailed Design:

Stage 5: Software Testing: Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. **What are the benefits of Software Testing?**

- Cost-Effective:
- Security:
- Product quality:
- Customer Satisfaction:

Stage 6: Software Maintenance: Software Maintenance is the process of modifying a software product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software applications after delivery to correct faults and to improve performance.

Difference between Traditional and Agile Software Development:

Traditional Software Development

Agile Software Development

It is used to develop the simple software. It is used to develop the complicated software.

In this methodology, testing is done once In this methodology, testing and development
the development phase is totally completed. Processes are performed concurrently.

It provides less security.

It provides high security.

It provides less functionality in the software. It provides all the functionality needed by the users.

It is basically used by freshers.

It is used by professionals.

Development cost is less using this

Development cost is high using this methodology.

Methodology.

It majorly consists of five phases.

It consists only three phases.

Benefits of traditional methodology

- Clearly defined objectives
- Clear documentation
- More accountability

Benefits of Agile project management

- Flexible prioritization
- Early and predictable delivery
- Predictable costs and schedules
- Improves quality
- More transparency
- High security
- **What is Agile?**
- Agile is a set of techniques followed by a team to administer a project or plan by dividing it into various stages with continuous collaboration with customers. There is constant monitoring at every phase of the software development of the project. The agile methodology advantages are that both the development plus testing actions are parallel and synchronized, unlike the conventional waterfall methodology.
- Agile is a project management methodology that breaks down larger projects into smaller, manageable chunks known as iterations. Contrary to Waterfall project management, agile is strictly sequenced. With agile, developers, designers, and business people are simultaneously working together.
- **Key Agile Concepts**
- **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.”
- **Daily Meeting:**
- **Team:**
- **Incremental Development:** Nearly all Agile teams favor an incremental development strategy
- **Iterative Development:** Agile projects are iterative insofar as they intentionally allow for “repeating” software development activities,
- **Milestone Retrospective:** Once a project is at the end, all of the team’s permanent members (not just the developers) invests from one to three days in a detailed analysis of the project’s significant events. (see more)

12 Principles are based on the Agile Manifesto.

- 1 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- 2 Welcome changing requirements, even late in development.
- 3 Deliver working software frequently, from a couple of weeks to a couple of months.
- 4 Business people and developers must work together daily throughout the project.
- 5 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 6 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 7 Working software is the primary measure of progress.
- 11 The best architectures, requirements, and designs emerge from self-organizing teams.
- 12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

TYPES: There are various types of agile methodology

2. **Kanban:** Kanban is a popular framework which is used to implement agile software development. The work items are represented in a Kanban board visually, allowing team members to see the state of every piece of work at any time. The Kanban board is the agile project management tool that designed the necessary visualized work, limited work-in-progress, and maximizes flow (or efficiency). It uses cards, columns, and provides continuous improvement to help technology and service teams who commit the right amount of work and get it done
3. **Scrum :** One of the most popular agile methodology examples is the agile scrum development methodology, which is depicted by various cycles of development. Similar to Kanban, Scrum breaks down the development phases into stages or cycles called 'sprints'. The development time for each sprint is maximized and dedicated, thereby managing only one sprint at a time.

Scrum and agile methodologies focus on continuous deliverables, and thus this method lets designers adjust priorities to ensure that any incomplete or overdue sprints get more attention.

1. **Extreme Programming (XP) :** Extreme Programming (XP) is a methodology that emphasizes teamwork, communication, and feedback. It focuses on constant development and customer satisfaction. Similar to scrum, this method also uses sprints or short development cycles. This is developed by a team to create a productive and highly efficient environment.

Extreme Programming technique is very supportive in a situation of constant and varying demands from the customers. It motivates the developers to accept changes in the customer's demands

2. **Crystal** Crystal is a group of smaller agile development methodologies comprising of Crystal Yellow, Crystal Clear, Crystal Red, Crystal Orange, and more. Each has its peculiar and exclusive framework that is characterized by factors such as system criticality, team size, and project priorities. Depending on the nature of the project or system criticality such as Comfort, Essential Money, Discretionary Money, and Life, the kind of crystal agile methodology is chosen.
3. **Dynamic Systems Development Method (DSDM):** To address the need for a standard industry charter for the swift delivery of software, the Dynamic Systems Development Method (DSDM) was developed. DSDM gives a comprehensive structure that is defined and modified to create a plan, execute, manage, and scale the procedure of software development. the DSDM believes that modifications to the project are always expected, and quality with timely delivery must never be negotiated.
4. **Feature Driven Development (FDD):** Several industry-recognized best practices are incorporated into this iterative, customer-centric, and incremental agile method. Its primary goal is to consistently produce working software in a timely fashion.

Lean Software Development: This agile methodology is based on seven principles:

- Deleting what doesn't matter-
 - Quality development-
 - Knowledge creation-
 - Delivery promptly- Providing value to the customer as quickly as possible
 - Respecting the team-
1. **Scaled Agile Framework (SAF):** A set of workflow and organizational patterns for implementing agile practices at an enterprise scale is known as the Scaled Agile Framework (SAF). Adopting SAF allows you to take advantage of a framework that is relatively light while still maintaining the centralized decision-making required at the enterprise level for software development efficiency. To put it another way, SAF takes the agile philosophy and applies it to software executives who are tasked with addressing more strategic issues.

Agile Ethics-

Agile ethics is the explicit application of agile methods to ethical assessment, adaptation, and learning that allows for a team to mature its practices. It employs agile methods to tackle ethical challenges.

Agile ethics requires four things:

- Decentralization of critical, ethical thinking in an organization or team
- Iterative development of process
- Dedicated staff time to manage the process
- Inclusion of diverse and external voices

Agile Ethics Job Description

In my experience, the team members most devoted to supporting ethical practice are either not embedded in the process or practice of inventing or are engaged in their ethics practices because they are particularly interested, not because they have been tasked with or supported to carry out the work. For an agile ethics process to work, some staff time has to be devoted explicitly to it.

Introduction to Scrum Scrum is an agile software development process to manage software projects. Scrum is based on three simple principles: visible progress, constant inspection, and adaptation. With Scrum, teams use an empirical approach to adapt to changing requirements and priorities. Teams using Scrum focus on delivering working software to their customers on a frequent basis.

The Scrum Team

There are three main roles in a Scrum team: Scrum Master, Product Owner, and Development Team.

The **ScrumMaster** role is somewhat similar to a project manager in traditional software development. The ScrumMaster helps the team follow the Scrum principles Unlike traditional project manager roles, the ScrumMaster does not assign individual tasks to the developers or QA but rather the development team self-organizes on each iteration and determines who will do each task. The ScrumMaster must be available at all times ensure that the team has all required resources for the job.

The **Product Owner** is a representative from the business side who determines the priorities of the items to be developed. The Product Owner must be either on site or highly available to the team at all times to answer questions and clarify unclear requirements.

The **Development Team** in Scrum can be composed of developers, and business analysts. In a Scrum team the development team provides estimates and does the work. However, the development team does not select what features should be developed-that is the job of the Product Owner. Once an iteration starts, the development team self-organizes and decides who will do each item and what is the best way to approach it. During each iteration, developers, and business analysts work on a daily basis to make sure the work is properly designed, coded, tested, fixed, and potentially shippable at the end of the iteration.

Process Overview

Scrum projects are managed in short iterations (Sprints) where the team works on the most important items of the project, as defined by the Product Owner, and delivers potentially shippable code at the end of each iteration. Everything in Scrum revolves around Sprints.

Sprint Planning

Before a Sprint starts, the Product Owner (with the help of the ScrumMaster) reviews the list of all new features, enhancements, change requests, and bug reports accumulated over time to decide which ones are most important at the time. If this is a new project, the list includes the features that the new system must provide. The entire list of items is called Product Backlog and each item must include a description of what is requested, the priority for the business. It is the job of the Product Owner to make sure this list is always up to date.

Sprint

A Sprint is an iteration in which the development team (developers and QA) works on the Sprint Backlog. During this time, the Product Owner must be available to answer questions and provide feedback on features as they start evolving. At the end of the Sprint, the development team has the responsibility of providing potentially shippable code.. Scrum recommends that the team gives a demonstration to the Product Owner of the software with the new features at the end of the Sprint.

Daily Meeting

During the Sprint, team members get together on a daily basis to report status to the team. The Scrum daily meeting-or daily scrum-is a short meeting time boxed at 15 minutes in which each participant answers three questions:

- What did you do since the last daily meeting?
- What will you do from now until the next daily meeting?
- Are there any roadblocks in your way?

Why Scrum Works

Scrum's practices and guidelines work because they address the most important component of software development: people and their interactions. Scrum provides a framework for the entire team (Product Owner, Scrum Master, developers, quality assurance, and business analysts to collaborate, and work with each other on a regular basis.

Progress Tracking

Teams using Scrum have several ways to track the status of the project. One is that at the end of the Sprint the team must have potentially shippable code. Having teams giving demonstrations to the Product Owner of the new features at the end of the Sprint is a great way to encourage this goal. At the end of the Sprint the software is potentially ready to be shipped or not

Agile Scrum methodology phases

The scrum models have 5 steps also called phases in scrum.

Step 1: Product Backlog Creation

In this step, the large items and functional details are transformed into epics and user stories. The user stories are transformed from large items and are smaller which can be put in the product backlog. The epics can also be included in the product backlog but cannot be included in the sprint backlog without converting it to a user story.

A typical user story example is as an admin I want to add, modify, and delete the tasks for the users in the website.

The following required fields in the user stories are necessary.

- The user stories significance
- The initial estimate made during the meeting
- Demo of how to make the user stories into tasks
- Know what is included in the product backlog

Step 2: Sprint planning and creating backlog

The sprint duration is very important so that the user stories are as small as possible. The typical average sprint duration lasts about 2 weeks. If the sprint duration is small than the advantage is that more customer feedback can be received and most of the errors and bugs can be addressed earlier. The next stage is to do the sprint backlog creation for which the scrum team must select the important user stories and make them into smaller tasks. They need to plan on how to get the task completed. Also, one important thing is to prioritize the necessary tasks.

Step 3: Working on sprint

The actual user stories are moved as small tasks in the sprint backlog where the actual work starts. This is where the realization of the software application for example the website development begins.

To begin with, a task board also called a Kanban board is made with a lot of cards is used. The cards specify the details about the tasks such as assignee, work details, due date or the time duration, etc. The task board consists of the following columns "Product backlog or the User stories", and the "To Do" lists, "Work In Progress" and then "Testing" and "Work Done" columns.

The cards can be moved from the left to right in the order of preference and based on the completion.

In this step the scrum meetings are important as it is done to track the progress status and who is doing what status. The burndown chart is the output of this meeting as it provides the details

about how many tasks are completed and if there any issues or problems in completing the tasks in the "Work In Progress" columns.

Step 4: Testing and Product Demonstration

Every sprint that is completed must be demonstrated to the customer for his acceptance and his viewpoint on the complete solution

Step 5: Retrospective and the next sprint planning

The result of this step is to discuss what went well and what can be improved for the next level. Also, you need to discuss the lessons learned. Then the next sprint planning has to be commenced based on the knowledge that we have for the current processes and past projects.

Final Takeaway

To know more and implement the scrum phases in real-time take up the CSM certification or the Scrum Master Certification online at StarAgile institute.

Agile Estimation

Agile estimation is about evaluating the effort required to complete each work item listed in the prioritized backlog, which, in turn, helps improve sprint planning. **What is Estimation in Agile?** Agile estimation is the process for estimating the effort required to complete a prioritized task in the product backlog. This effort is usually measured with respect to the time it will take to complete that task, which, in turn, leads to accurate sprint planning.

What is a Sprint? A sprint is a time-boxed interval that defines the time allocated to complete a task.

Note: No matter how accurately a business estimates the effort required to complete a user story in Agile, an estimate is still an estimate. Do not strive to achieve perfect accuracy because requirements can change at any time.

Agile teams also make estimations with reference to story points. A story point is used in Agile Development projects to estimate the difficulty of implementing a given user story. This is measured in relative units assigned to different user stories that require estimation.

In a nutshell, a story point is a number that helps estimate the difficulty of building a user story successfully. This difficulty could be anything related to the complexities, risks, and efforts involved.

Why Run Agile Estimations?

- Agile estimations are essential for:
- Making teams accountable for deliverables
- Inducing discipline across the Agile team
- Predicting the approximate time it will take to finish a project
- Enabling better sprint management

- Improving team productivity

Why do Teams Estimate in Agile?

Overestimating and underestimating are both common for Agile development teams which leads to varying development and launch times. Though the process is complicated, considering Agile estimation in the initial stages can assist with accurate user story estimations and helps the team stick to the timely deliverables.

Some of the to-the-point benefits of Agile Estimation techniques include:

1. Improved Decision-Making

With accurate, agile estimation, the development team will be able to conduct effective backlog grooming sessions, which will further help in precise sprint planning. When they make informed decisions and plan well, their user story delivery time will improve.

2. Better Coordination

Let's say that the estimated effort for user story A is two weeks. On the other hand, the estimation effort for user story B is four weeks. Now, suppose both the user stories depend on each other and are connected. In that case, the team needs to prioritize work so that both user stories get completed simultaneously, thus leading to better coordination among teams.

3. Better Risk Management

Software projects often suffer from exceeding budgets and timelines. To lessen this risk, Agile project estimation is an ideal solution. Agile product estimation helps estimate story points and stick to budgets, estimates, and the project's scope. The more accurate the estimates, the better the chances of on-time, quality delivery.

The in-detail process includes:

1. Conduct Stakeholder Interviews

- Business Requirements Document (BRD): defines the end-goal of the project
- Functional Requirements Document (FRD): defines the features required to achieve the end-goal.

2. Define High-Level Product Backlog

3. Understand the Client and its Potential Customers

.

4. Prioritize Requirements

5. Prepare the Minimum Viable Product (MVP) Backlog

Based on the prioritization activity, the BA assembles the requirements as 'must-haves' to the backlog and sections them as the requirements for the MVP Development.

6. Estimate the Project Cost and Timeline

Product Backlog

What is a product backlog?

A product backlog is a prioritized list of work for the development team that is derived from the roadmap and its requirements. The most important items are shown at the top of the product backlog so the team knows what to deliver first. the development team pulls work from the product backlog as there is capacity for it, either continually (kanban) or by iteration (scrum).

Start with the two "R"s

A team's roadmap and requirements provide the foundation for the product backlog. Roadmap initiatives break down into several epics, and each epic will have several requirements and user stories.

The product owner then organizes each of the user stories into a single list for the development team. The product owner may choose to deliver a complete epic first (left). Or, it may be more important to the program to test booking a discounted flight which requires stories from several epics (right). See both examples below.

What may influence a product owner's prioritization?

- Customer priority
- Urgency of getting feedback
- Relative implementation difficulty
- Symbiotic relationships between work items (e.g. B is easier if we do A first)

While the product owner is tasked with prioritizing the backlog, it's not done in a vacuum. Effective product owners seek input and feedback from customers, designers, and the development team to optimize everyone's workload and the product delivery.

Sprint Backlog

What is a Sprint Backlog?

A sprint backlog is the set of items that a cross-functional product team selects from its product backlog to work on during the upcoming sprint.

Typically the team will agree on these items during its sprint planning session. In fact, the sprint backlog represents the primary output of sprint planning.

Sprint Backlog vs. Product Backlog: What's the Difference?

The product backlog is the comprehensive list of product-related tasks that, at any given time, a sprint backlog is a much shorter list pulled from the items on the product backlog.

The distinction between sprint backlogs and product backlogs, and how the two work together:

1. Sprint backlog items should be taken directly from the product backlog.
2. While the product backlog can be changed frequently at any time, according to the always-changing realities in an organization the sprint backlog should remain as fixed as possible throughout the duration of the sprint.
3. The top items on a well-groomed, prioritized product backlog will often represent the upcoming sprint backlog.
5. If the team is unable to complete certain sprint backlog items by the end of the sprint, the team might choose to add those unfinished jobs either to the next sprint or to the product backlog to be addressed again in the future.

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal.

Commitment: Sprint Goal

The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the Developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

The Sprint Goal is created during the Sprint Planning event and then added to the Sprint Backlog. As the Developers work during the Sprint, they keep the Sprint Goal in mind. If the work turns out to be different than they expected, they collaborate with the Product Owner to negotiate the scope of the Sprint Backlog within the Sprint without affecting the Sprint Goal.

Increment

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable.

Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value.

Work cannot be considered part of an Increment unless it meets the Definition of Done.

Search all Resources related to Backlogs.

Building a Requirement Document

Initial Stages of Building a Requirement Document

Good Requirements:

- User Stories
- User Acceptance Tests
- Workflow
- Requirements (Details)
- Wireframes

Without any one of the aforementioned sections, the requirements start to lose value. Each section brings a lot to the table, and too often they are judged as a “waste of time.” Much like pair programming, having each of the sections for your requirements actually creates a synergy, as the development process is sped up. (Sorry for the consultant buzzword, but it does)

The rough outline of the structure is as follows:

1. Define document properties

This helps get everyone on the same page in terms of who is responsible for what and communicating our target release goals. We use these properties to also report and quickly access previous requirements on an index page showing all the requirements:

2. Communicate the overall goals

Be short and straight to the point to explain what you are trying to achieve.

3. Background and strategic fit

Why are we doing this? How does this fit into our overall vision of our product, company or epic?

4. Assumptions

List the technical, business or user assumptions you might be making. This is very helpful in providing context as someone reads your user stories.

5. User Stories

We list the user stories involved here typically in a tabular format. Where possible, try to include context. Link to customer interviews, take screenshots of what you've seen. Embed audio or multimedia content of your customers. Provide as much detail as you need to tell the story. How will we know we've succeeded? We often note success metrics for each story in this section as well.

6. User interaction and design

Quite often filled out after we discuss the problems and flesh them out, we keep the designs embedded or linked from the same page.

7. Questions

As we are thinking about our problems, we often uncover new questions that need answering. So we typically have a table of the "things we need to decide on or do further research". This table typically grows towards the definition phase and as we start implementing we find ourselves coming back to it and updating it as decisions are made.

BURNDOWN CHART

WHAT IS A BURNDOWN CHART?

A burndown chart shows the amount of work that has been completed sprint, and the total work remaining. Burndown charts are useful because they provide insight into how the team works. For example:

- If you notice that the team consistently finishes work early, this might be a sign that they aren't committing to enough work
- If they consistently miss their forecast, this might be a sign that they've committed to too much work.

WHAT IS AN ESTIMATION STATISTIC?

The estimation statistic is the unit of measurement your team will use to estimate work. It's used to calculate team velocity. For each sprint, the velocity is the sum of the Estimation Statistic for completed stories. If your team is consistent with its velocity, you can use it to determine how much work they can take on in each sprint.

WHICH ESTIMATION STATISTIC SHOULD WE USE? Traditionally, software teams estimated their work in a time format using days, weeks, and months. However, many agile teams have transitioned to story points. We recommend using story points.

Sprint burndown chart

This report shows the amount of work to be done in a sprint. It can be used to track the total work remaining in the sprint. By tracking the remaining work throughout the sprint, a team can manage its progress, and respond to trends accordingly. For example, if the burn down chart shows that the team may not reach the sprint goal, then they can take the necessary actions to stay on track.

Understanding the sprint burn down chart

- Estimation statistic:
- Remaining values: The red line represents the total amount of work left in the sprint, according to your team's estimates.
- Guideline: The grey line shows an approximation of where your team should be,

assuming linear progress. If the red line is below this line, congratulations - your team's on track to completing all their work by the end of the sprint.

Epic burndown chart

This report shows you how your team is progressing against the work for an epic. Here are some of the ways that you could use an epic burndown chart:

- See how quickly your team is working through the epic.
- See how work added/removed during the sprint has affected your team's overall progress.
- ☐ Predict how many sprints it will take to complete the work for an epic,

Sprint Planning

Definition : Sprint planning is an event in the Scrum framework where the team determines the product backlog items they will work on during that sprint and discusses their initial plan for completing those product backlog items.

Teams may find it helpful to establish a sprint goal and use that as the basis by which they determine which product backlog items they work on during that sprint.

Who is involved in Sprint Planning : Sprint planning typically involves the entire team.

A product owner identifies the candidate product backlog items and their relative priorities, proposes a sprint goal.

The team members determine how many of the product backlog items they forecast they will be able to complete and determine how they will deliver those product backlog items.

When does Sprint Planning take place?

Sprint planning occurs on the first day of a new sprint.

The event should occur after the sprint review so that any output from those discussions can be considered when planning for the new sprint.

How is Sprint Planning Structured?

Sprint planning is typically split into two parts:

Part 1 – Scope

The team selects which items from a prioritized list of ready product backlog items they forecast they will be able to complete during the sprint.

Here's a sample agenda for the first part of sprint planning:

- ☐ What is the goal for this sprint?
- What product backlog items are ready and contribute toward the sprint goal?
- Who is available for this sprint?
- How confident does the team feel that they'll be able to meet the sprint goal.

Part 2 – Plan

The team discusses in more detail how they will deliver the selected product backlog items. This may include identifying tasks for the product backlog items,

Expected Benefits

The main benefit of sprint planning is that it allows a team to start a new sprint with a shared understanding of what they will work on for that sprint, as well as an initial plan for how they approach that work.

Common Pitfalls

Sprint planning can become ineffective when your team does not have a properly refined product backlog from which to draw product backlog items. If you find yourself in this situation you inevitably have to spend time during sprint planning developing a better understanding of the product backlog items

When Applicable

Sprint planning is typically used when your team is following the Sprint framework,

Potential Costs

Sprint planning can become costly when your team spends an inordinate amount of time planning compared to the act of delivering the product backlog items.

The Scrum Guide identifies a maximum of 8 hours for a month long sprint.

What Details Are Prioritized During a Sprint Planning Meeting?

The objective of sprint planning is to work out the key details regarding the team's planned work during the next sprint. With that in mind, the sprint team should plan to address at least the following issues during this meeting.

- Decide on the team's overall strategic objective for the next sprint.
 - Review the product backlog and discuss which items belong on the next sprint backlog and why.
 - Call for a team consensus on the proposed sprint goal and backlog items (led by the scrum master).
 - Discuss team capacity.
 - Discuss known issues that could disrupt or slow progress on the sprint backlog.
 - Assign the new sprint backlog's tasks, according to skill sets, capacity, and other relevant criteria.
 - Confirm the timeframe of the upcoming sprint.
- Open the meeting to sprint-related questions.

Sprint Retrospective

What is a Sprint Retrospective?

The sprint retrospective is a recurring meeting held at the end of a sprint used to discuss what went well during the previous sprint cycle and what can be improved for the next sprint.

The Sprint Retrospective occurs after the Sprint Review and prior to the next Sprint Planning. This is at most a three-hour meeting for one-month Sprints. The retrospective session is attended by all – the product owner, scrum master, development team members, and optionally with the stakeholders.

5 Steps for Conducting Sprint Retrospective Meeting

- Set the stage – Set the goal; Give people time to “arrive” and get into the right mood
- Gather data – Help everyone remember; Create a shared pool of information (everybody sees the world differently)
- Generate insight – Why did things happen the way they did?; Identify patterns; See the big picture
- Decide what to do – Pick a few issues to work on and create concrete action plans of how you'll address them
- Close the retrospective – Clarify follow-up; Appreciations; Clear end; How could the retrospectives improve?

Retrospective Meeting Questions

During the Sprint Retrospective, the team discusses:

- What went well in the Sprint?
- What went wrong in the Sprint?
- What We had Learn in the Sprint?
- What should we do differently in the next sprint?

1. What went well in the Sprint?

- What motivated us to do it?
- What did we do differently to make it a success?
- Which training,

skill, or knowledge contributed to the difference?

- Which strong points of your team that made it happen?
- What is the team member's contribution that helped you accomplish it?
- How did we achieve it?

2. What went wrong in the Sprint?

3. What We had Learn in the Sprint?

4. What Should We do Differently in the Next Sprint?

- How can the strength of the individual be utilized to resolve the issue?
- What should be done often to prevent the issue from arising again?
- Identify the 1 thing to be changed and explain how you could change it?
- What strategies will work to complete the job?

Corrective actions

- What will you do in the upcoming Sprint to complete this action?
- How will you do it to make it a success?
- What additional support do you require?
- How will you let me know that you completed it?
- What will you do next after accomplishing this during the Sprint?

Daily Scrum

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

Inspection and Adaption at the Heart of Scrum

The Developers use the Daily Scrum to inspect progress toward the Sprint Goal and to inspect how progress is trending toward completing the work in the Sprint Backlog. The Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal. Every day, the Developers should understand how it intends to work together as a self-organizing team to accomplish the Sprint Goal by the end of the Sprint. Daily

The structure of the meeting is set by the Developers and can be conducted in different ways if it focuses on progress toward the Sprint Goal.

During the daily scrum, each team member answers the following three questions:

- What did you do yesterday?
- What will you do today?
- Are there any impediments in your way

Guideline for Daily Standup meeting

Here are some useful guidelines for a effective daily standup meeting:

- The Meeting should be held at fixed locations at fixed times, and should not be late, leave early, or absent.
- The so-called standup meeting, as the name suggests, is standing up and meeting, should not sit, or even lie down. Standing up in a meeting can make the meeting faster and more efficient, and it is guaranteed to be completed within 15 minutes.
- Everyone should state their work status in a concise and appropriate manner, and do not go into details.
- Everyone in the meeting takes turns to speak to each member of the team about progress status . It aims to enhance communication and maximize collaboration of the entire team.
- One person at a time taking turns to speak and each person speaks only once. When someone speaks, other members should pay attention to listening,

Scrum Product Owner

What is a Scrum Product Owner?

In the Scrum methodology, the Scrum Product Owner is usually a project's key stakeholder— typically someone from marketing or product management, or the lead user of a system. They have a deep understanding of users, the marketplace, competitors, and trends.

One of their key responsibilities is managing the product backlog—the prioritized features list for the product. The Scrum Product Owner prioritizes work during the sprint planning meeting and motivates the team with clear goals, answering any questions.

Other key responsibilities and roles of a certified Scrum Product Owner include:

- Defining the vision: the product development team, communicating the goals and vision to customers, business managers, and the development team. They may also provide a product road map to the team
- Prioritizing the product backlog: After creating a product backlog list, a Scrum Product Owner prioritizes items according to business objectives and overall strategy
- Taking an overview of development stages: After the Scrum Product Owner sets the vision, strategy, and product priorities, they oversee the development of the product through all its stages, including the planning, refinement, review, and sprint for each event
- Handling communications: Scrum Product Owners must be the primary communicators between stakeholders and development teams..
- Knowing what the client needs: By thoroughly understanding client requirements and anticipating what they want, a certified Scrum Product Owner can better manage the development process, and find good solutions.

Evaluating progress: Since the Scrum Product Owner is accountable for the final product and each stage of its development, they must take a leading role in evaluating the product progress through each iteration.

What makes a good Scrum Product Owner?

They need to be effective communicators, keeping touch with the development team and internal and external stakeholders.

They also need to have:

- Great teamwork skills
- Past success in releasing products and meeting business objectives
- Subject expertise in a particular product or market
- A proven record of influencing cross-functional teams

Scrum Master

What is a Scrum master?

A Scrum master is a facilitator who ensures that the Scrum team follows the processes that they agreed to follow. The Scrum master skillfully removes obstacles and distractions that may impede the team from meeting goals.

Scrum methodology is a subset of agile software development in which a development team sets goals to deliver new software features and functionality in well-defined, iterative cycles. Each iteration delivers small, but significant, parts of an overall project every two to four weeks.

5 key Scrum master responsibilities

1. Coach team members

The Scrum master makes sure that team members are well trained and understand Agile processes. The Scrum master also ensures that team members know their respective roles

2. Host daily stand-up meetings

Daily Scrum, or stand-up, meetings last no longer than 15 minutes and give each team member the opportunity to answer these questions:

- What did you do yesterday?
- What will you do today?
- What is impeding your progress?

Team members may be asked to estimate how much time they will need to complete specific tasks. The Scrum master hosts this meeting, tracks team progress, and notes any obstacles that may keep the team from completing tasks.

3. Assist the product owner with the product backlog

The product owner is responsible for creating and maintaining the product backlog. The product backlog is a list of work that the team needs to do. The Scrum master helps the product owner refine and maintain the product backlog using the information gathered from the daily stand-up meetings.

4. Remove roadblocks

To help the team stay focused on the tasks that need to be done during each iteration, the Scrum master looks for distractions and roadblocks that can impede progress

For example, if team members are being pulled into too many unimportant meetings, the Scrum master can work with meeting organizers to determine who really needs to attend the meetings.

5. Teach Scrum practices and principles

To ensure that work does not slow down, a key Scrum master role is to act as a mentor and teacher to smoothly onboard new employees and new team members. As a teacher, the Scrum master helps new team members to understand the scope and vision of a product. The Scrum master teaches the team how to be self-organized and how to stay focused.

Develop excellent verbal and written communication skills

One of the most important Scrum master skills you will need is communication. As part of your Scrum master responsibilities, you will be the main communicator among project owners and other key stakeholders. If you do not have good verbal and written communication skills, or if you are uncomfortable interacting one-on-one or in groups, this may not be the job for you.

Consider Agile Scrum master certification

There are Scrum master certification courses available, and although some companies will consider certification as one of their Scrum master prerequisites, it's not always necessary. However, if you have a certificate, it may help to give you an edge over other applicants when applying for Scrum master jobs.

Successful Scrum master characteristics also include the ability to communicate effectively and organize, motivate, manage and improve processes. Whether you ultimately end up as a Scrum master, all of these skills can help you progress in just about any industry.

Scrum team

A development team is composed of the professionals who do the hands-on work of completing the tasks in a Scrum sprint. This means development team members can be computer engineers, designers, writers, data analysts.

Not all development team members will always have the same responsibilities. For example if you're updating a website, you might have a front-end engineer, UX designer, copywriter, and marketing professional all working on the same Scrum team. The responsibilities of a development team will also depend on the end goals of the Scrum team.

a development team can be tasked with the following:

- Help in sprint planning and goal setting
- Lend expertise to program, design, or improve products
- Use data to find best practices for development
- Test products and prototypes, plus other forms of quality assurance

Within the Scrum Framework all work delivered to the customer is done by dedicated Scrum Teams. A Scrum Team is a collection of individuals working together to deliver the requested and committed product increments.

Within the Scrum Framework all work delivered to the customer is done by dedicated Scrum Teams. To work effectively it is important for a Scrum Team that everyone within the team

- follows a common goal
- adheres to the same norms and rules
- shows respect to each other

Characteristics of a Scrum Team

Scrum Teams always have the following characteristics:

- Team members share the same norms and rules
- The Scrum team as a whole is accountable for the delivery
- The Scrum Team is empowered
- The Scrum Team is self-organizing
- The skills within the Scrum team are balanced
- A Scrum Team is small and has no sub-teams

The development team:

The development team are the people that do the work According to the Scrum Guide, the development team can be comprised of all kinds of people including designers, writers, programmers, etc.

The development team should be able to self-organize so they can make decisions to get work done. The development team can make decisions and deliver the fix for the problem at hand. Self-organization is about empowering the people closest to the work to do what's needed to solve the problem.

The development team's responsibilities include:

- Delivering the work through the sprint.
- To ensure transparency during the sprint they meet daily at the daily scrum. The daily scrum provides transparency to the work and provides a dedicated place for team members to seek help, talk about success and highlight issues and blockers.