TECHPROEDUCATION

SDLC 132 137 141-142

Dashboard / My Courses / Automation Engineer / Automation Engineer EN / B 132 133 QA ENG / SDLC 132 137 141-142 / Preclass / Preclass

Lesson menu

Software Development Methodologies
Software Development Team Roles
What is Software Development Life Cycle (SDLC)?

Preclass

Mark as done

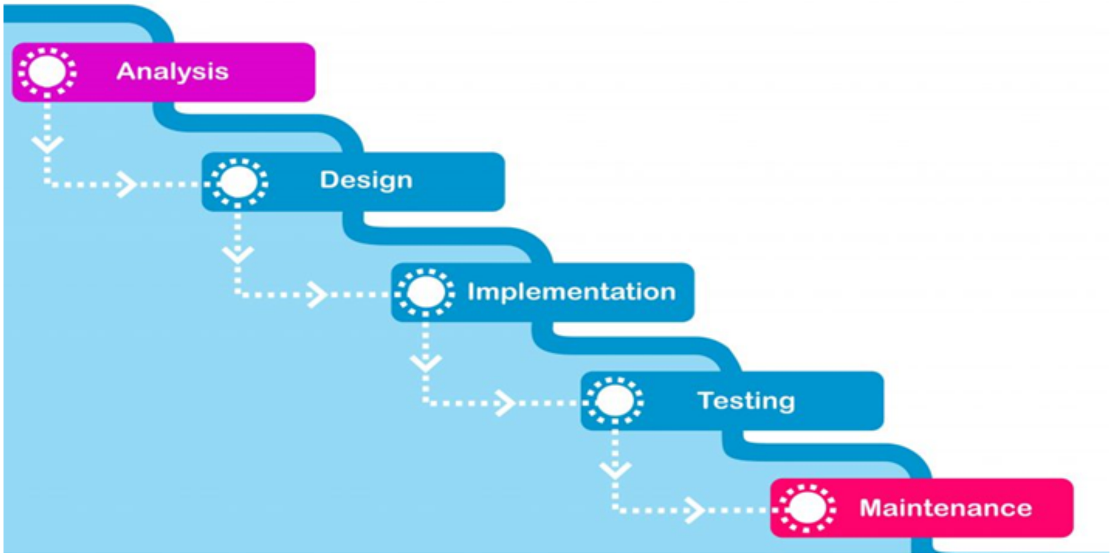
Software Development Methodologies

When it comes to software development, Waterfall is the most traditional and sequential choice. Although it's usually viewed as an "old school" or outdated method, it's helpful to understand the history and structure of Waterfall to better appreciate the flexibility of more modern methodologies. First created in 1970, Waterfall was one of the most prominent methodologies for several decades because of its plan-driven approach.

Waterfall requires plenty of structure and documentation up front. It is divided into self-contained stages or steps. The first stage is vital, requiring a full understanding by both developers and customers of the project's demands and scope before anything begins. The stages are relatively rigid and often follow this sequence: determine the project's requirements and scope, analyze those requirements, design, implement, test, deploy and finally, maintain.

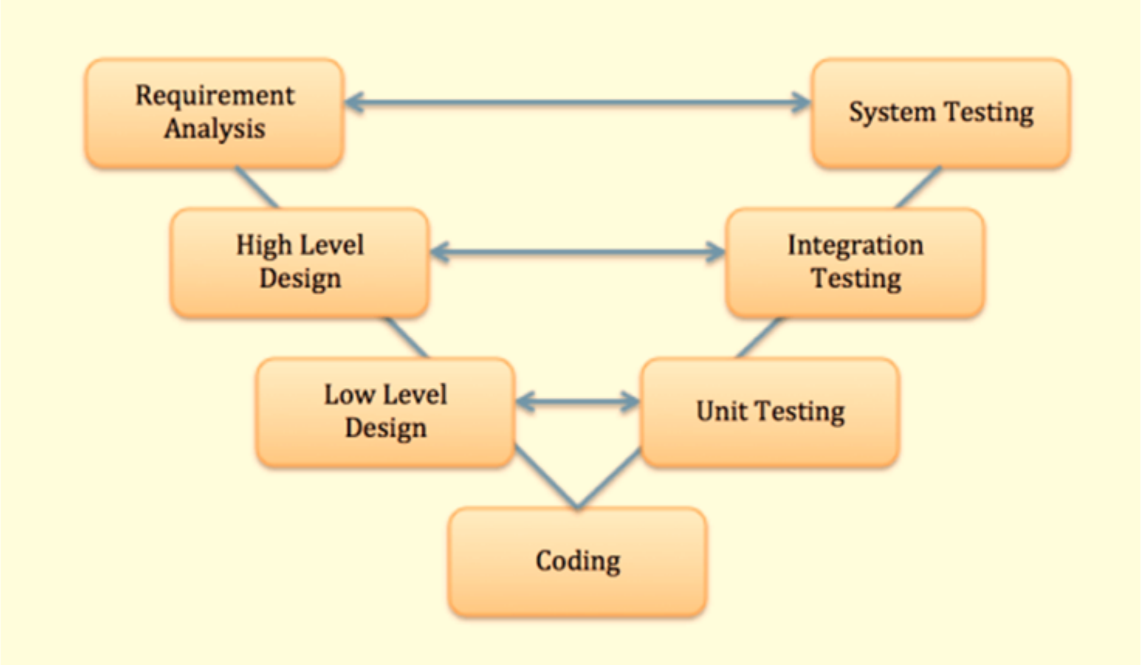
There's a lack of flexibility with this approach, meaning what is decided by the customer and developer at the beginning must be seen through. Should any changes need to be made or mistakes addressed toward the end stages, the Waterfall method generally requires a full restart.

Typically, one stage must be finished before the next can begin, which can help with organization and assignments. And because the full scope of the project is understood in advance, software progress can easily be measured. Waterfall is often utilized by large, plan-driven teams who have a very clear understanding of the project's scope;—however, development teams who don't operate in a vacuum will likely find better results with the flexibility and agility of more modern methodologies.

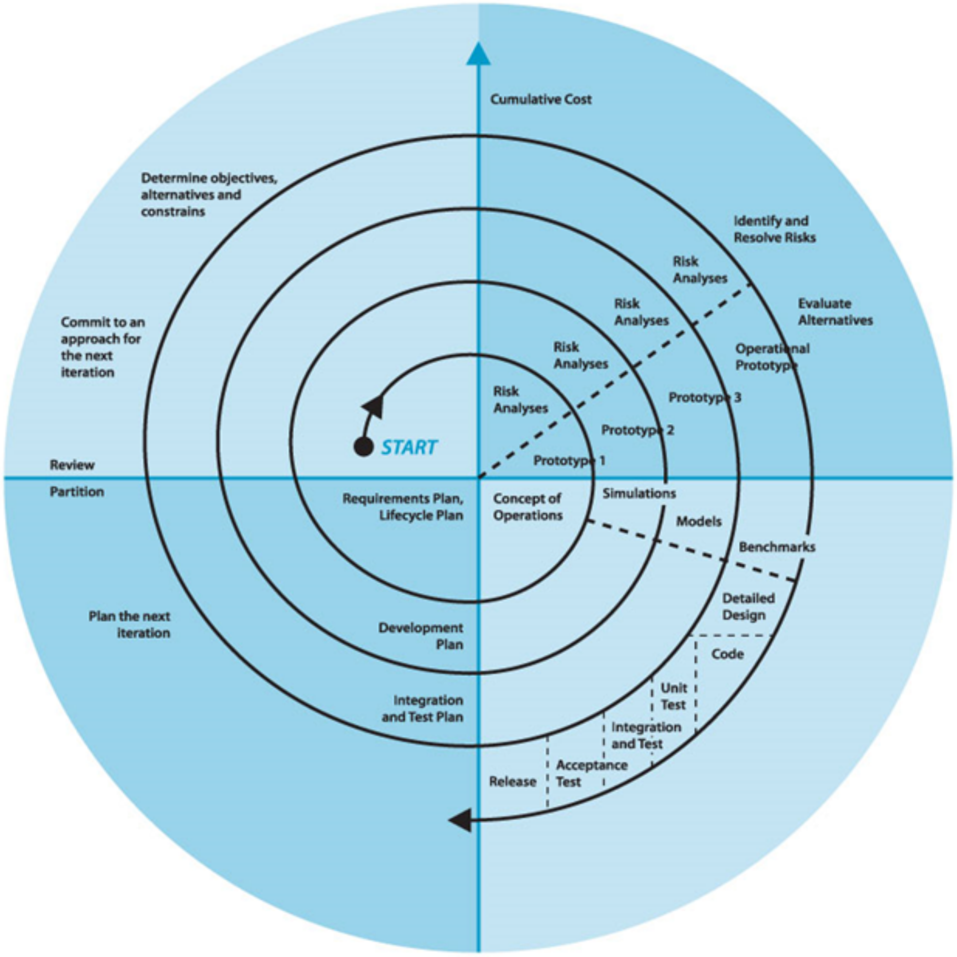


V Shape Model

V Shape Model is a highly disciplined SDLC model which has a testing phase parallel to each development phase. The V model is an extension of the waterfall model wherein software development and testing is executed in a sequential way. It is known as the Validation or Verification Model.



Spiral Model



The Spiral Model is a sophisticated model that focuses on the early identification and reduction of project risks. In this software development methodology, developers start on a small scale then explores the risks involved in the project, make a plan to handle the risks, and finally decides whether to take the next step of the project to do the next iteration of the spiral. The success of any Spiral Lifecycle Model depends on the reliable, attentive, and knowledgeable management of the project.

Advantages of Spiral Model

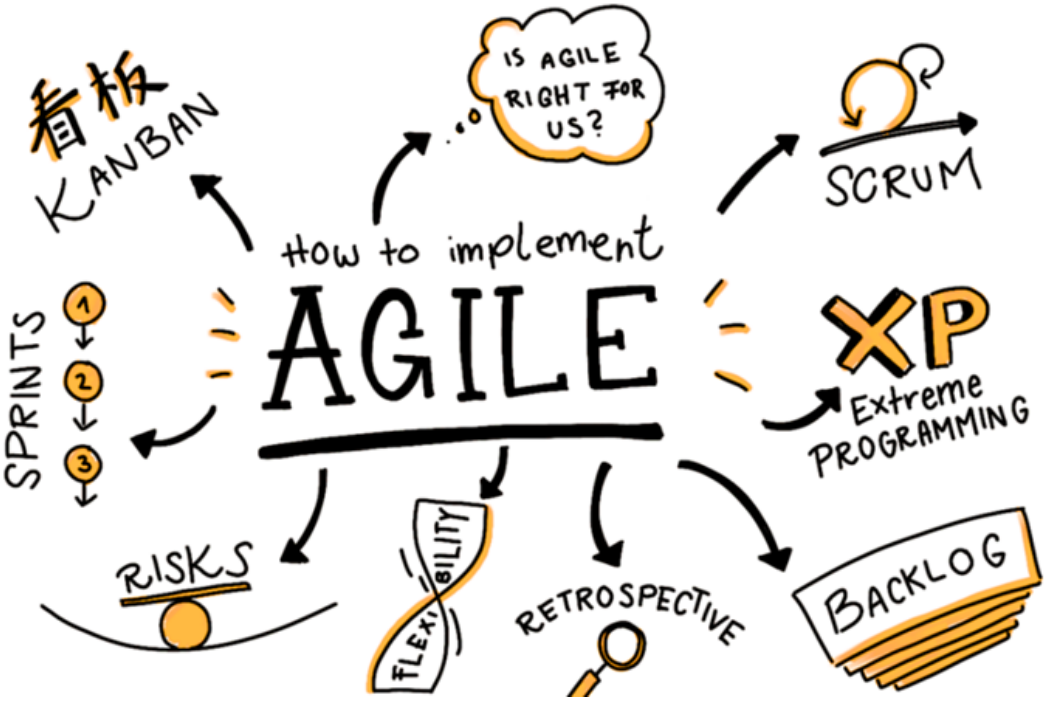
- The high amount of risk analysis being done hence, avoidance of possible risk is certainly reduced with this model.
- This model is good for large size and critical projects.
- In the spiral model, additional functionality can be added at a later date.
- Development is fast and features are added systematically in this model.
- It is more suited for high-risk projects, where business needs may differ from time to time basis.

Disadvantages of Spiral Model

- It is certainly a costly model to use in terms of development.
- The success of the entire project is dependent on the risk analysis phase thus, failure in this phase may damage the entire project.
- It is not appropriate for low-risk projects.

- The big risk of this methodology is that it may continue indefinitely and never finish.
- Documentation is more as it has intermediate phases.

Agile



The Agile methodology was developed as a response to growing frustrations with Waterfall and other highly structured, inflexible methodologies. This approach is designed to accommodate change and the need to produce software faster.

Agile values individuals and their relationships and interactions over tools; it features customer collaboration throughout the development process; it responds to change instead of following a set-in-stone plan; and it focuses on presenting working software, rather than documentation.

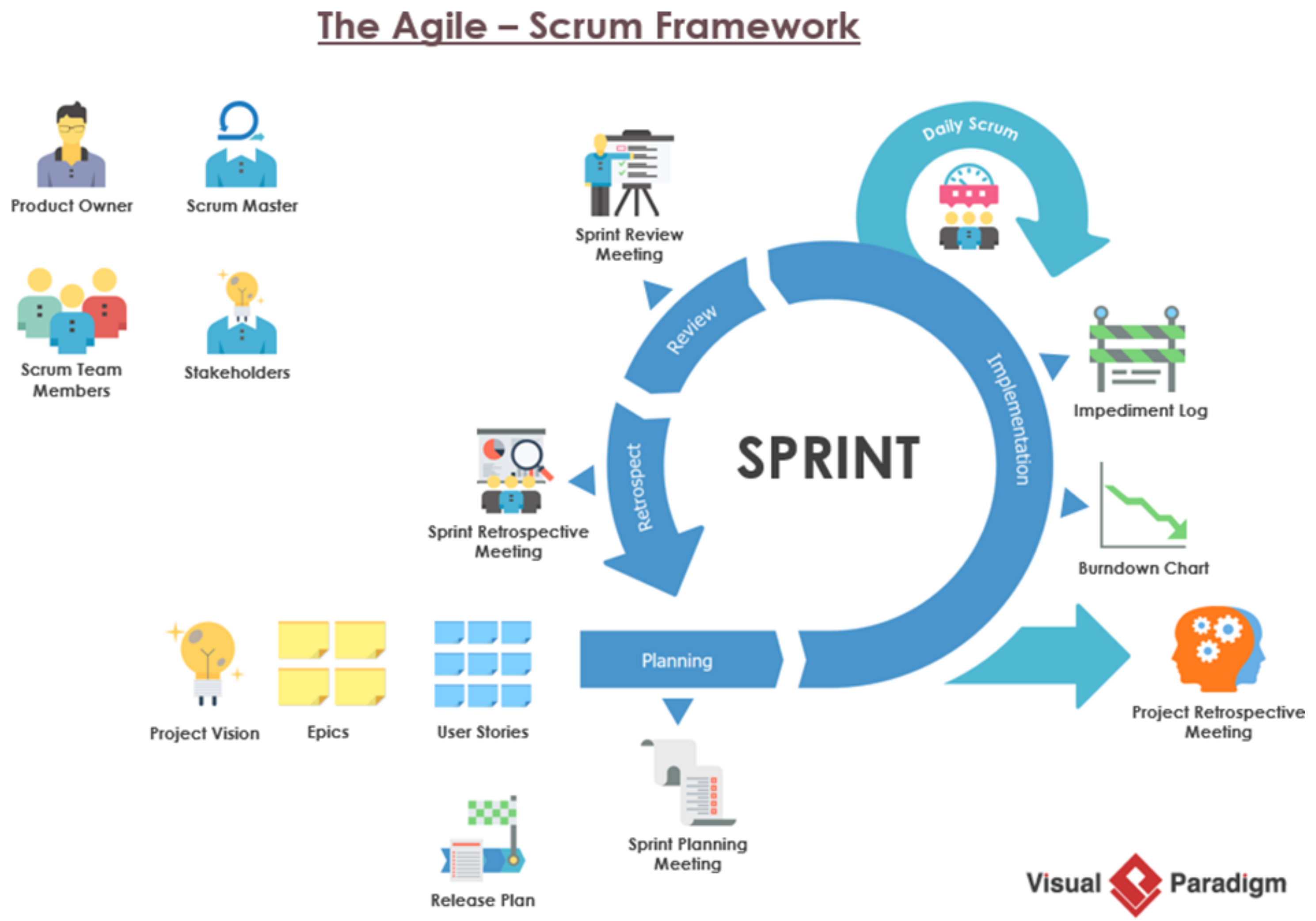
Unlike Waterfall, Agile is well equipped to handle the complexity and variability involved in development projects.Using the Agile approach, teams develop in short sprints or iterations, each of which includes a defined duration and list of deliverables, but in no particular order. During sprints, teams work towards the goal of delivering working software (or some other tangible, testable output).

Agile is collaboration-heavy, focusing on team strengths and efficiency, along with internal feedback from various departments and clients. Client satisfaction is the highest priority with the Agile approach, which teams achieve by continuously delivering working, tested, prioritized features.

Kanban and Scrum are the two most popular methods in Agile.

A Brief Overview of the Scrum Framework

Scrum is an agile development methodology used in the development of Software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project. The primary objective of Scrum is to satisfy the customer's need through an environment of transparency in communication, collective responsibility and continuous progress. The development starts from a general idea of what needs to be built, elaborating a list of characteristics ordered by priority (product backlog) that the owner of the product wants to obtain.

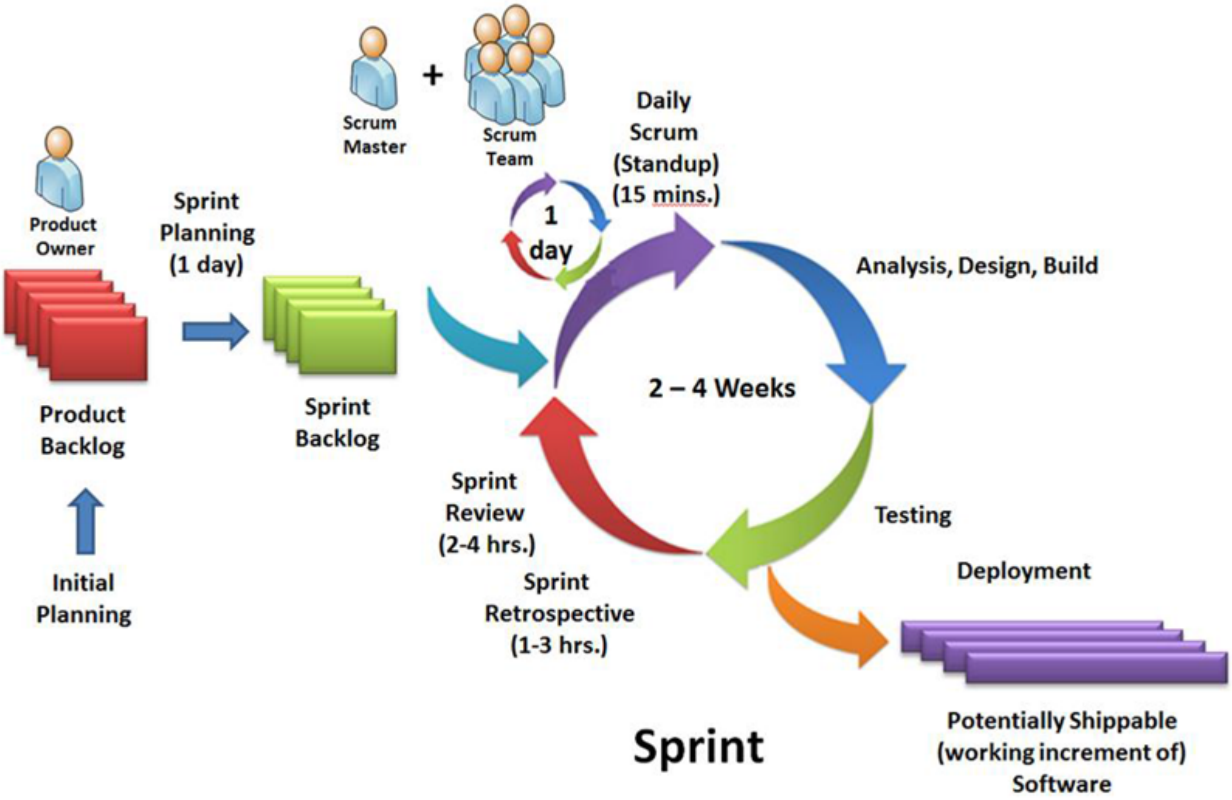


Scrum Methodology & Process

Scrum is precisely an evolution of Agile Management. Scrum methodology is based on a set of very defined practices and roles that must be involved during the software development process. It is a flexible methodology that rewards the application of the 12 agile principles in a context agreed by all the team members of the product.

Sprint

Sprint is the basic unit of work for a Scrum team. This is the main feature that marks the difference between Scrum and other models for agile development.



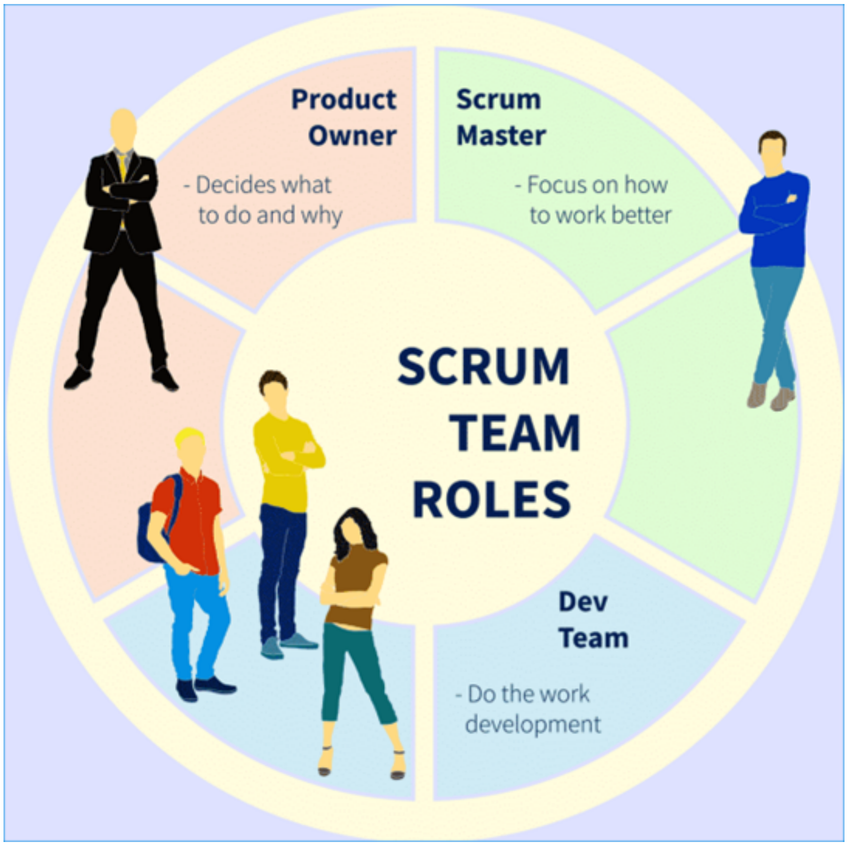
Scrum is executed in temporary blocks that are short and periodic, called Sprints, which usually range from 2 to 4 weeks, which is the term for feedback and reflection. Each Sprint is an entity in itself, that is, it provides a complete result, a variation of the final product that must be able to be delivered to the client with the least possible effort when requested.

The process has as a starting point, a list of objectives/ requirements that make up the project plan. It is the client of the project that prioritizes these objectives considering a balance of the value and the cost thereof, that is how the iterations and consequent deliveries are determined.

On the one hand the market demands quality, fast delivery at lower costs, for which a company must be very agile and flexible in the development of products, to achieve short development cycles that can meet the demand of customers without undermining the quality of the result. It is a very easy methodology to implement and very popular for the quick results it gets.

Scrum methodology is used mainly for software development, but other sectors are also taking advantage of its benefits by implementing this methodology in their organizational models such as sales, marketing, & HR teams etc.

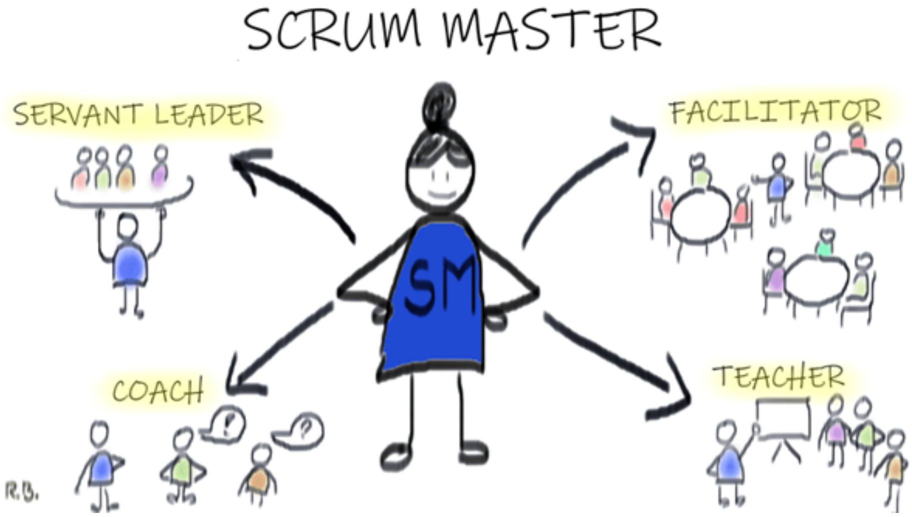
Different Roles in Scrum


Product Owner
- Decides what to do and why
Scrum Master
- Focus on how to work better
SCRUM TEAM ROLES
Dev Team
- Do the work development

In Scrum, the team focuses on building quality software. The owner of a Scrum project focuses on defining what are the characteristics that the product must have to build (what to build, what not and in what order) and to overcome any obstacle that could hinder the task of the development team.

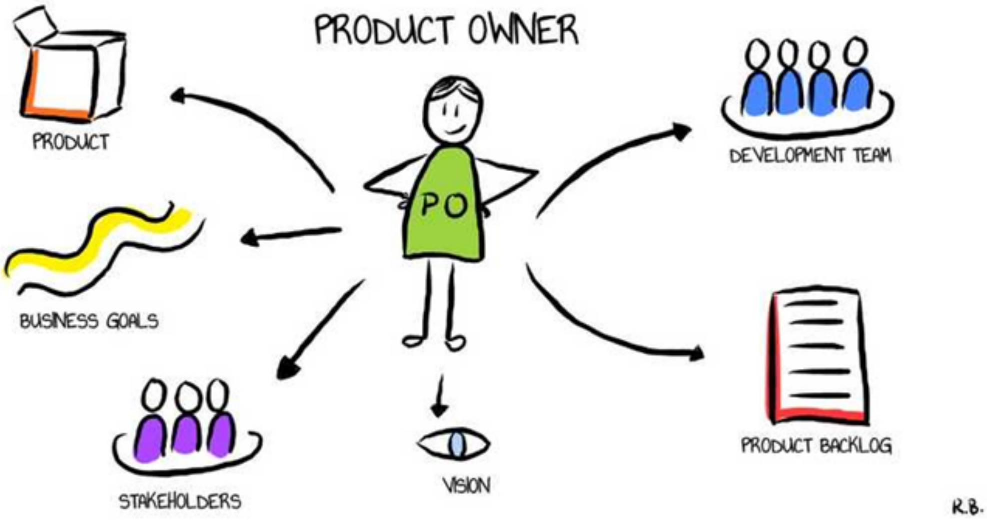
The Scrum team consists of the following roles:

Scrum master:

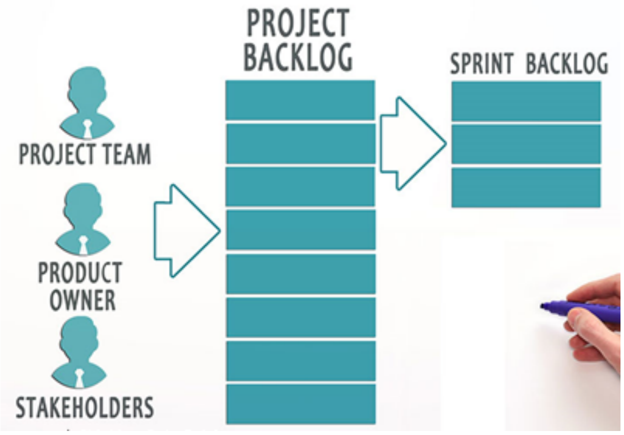


The person who leads the team guiding them to comply with the rules and processes of the methodology. Scrum master manages the reduction of impediments of the project and works with the Product Owner to maximize the ROI. The Scrum Master is in charge of keeping Scrum up to date, providing coaching, mentoring and training to the teams in case it needs it.

Product owner (PO):



Is the representative of the stakeholders and customers who use the software. They focus on the business part and is responsible for the ROI of the project. They Translate the vision of the project to the team, validate the benefits in stories to be incorporated into the Product Backlog and prioritize them on a regular basis.



Development Team:

A group of professionals with the necessary technical knowledge who develop the project jointly carrying out the stories they commit to at the start of each sprint.

Benefits of Scrum Methodology

Scrum has many advantages over other agile development methodologies. It is currently the most used and trusted framework of reference in the software industry. Below are some of the known benefits of Scrum:

Easily Scalable: Scrum processes are iterative and are handled within specific work periods, which makes it easier for the team to focus on definite functionalities for each period. This not only has the benefit of achieving better deliverables in line with the needs of the user, but also gives the ability to the teams to scale the modules in terms of functionality, design, scope and characteristics in an orderly, transparent and simple manner.

Compliance of expectations: The client establishes their expectations indicating the value that each requirement/ history of the project brings, the team estimates them and with this information the Product Owner establishes its priority. On a regular basis, in the sprint demos, the Product Owner verifies that the requirements have been met and transmits feedback to the team.

Flexible to changes: Quick reaction to changes in requirements generated by customer needs or market developments. The methodology is designed to adapt to the changing requirements that complex projects entail.

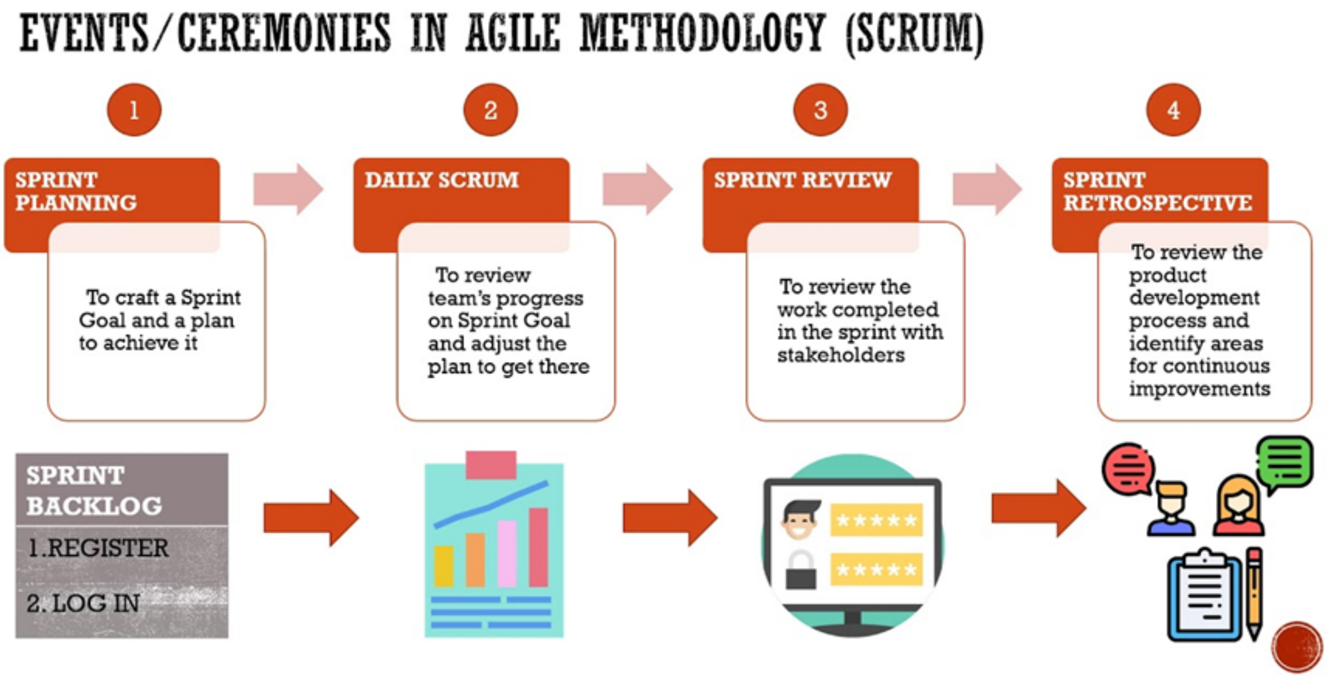
Time to Market reduction: The client can start using the most important functionalities of the project before the product is completely ready.

Higher software quality: The working method and the need to obtain a functional version after each iteration, helps to obtain a higher quality software.

Timely Prediction: Using this methodology, we know the average speed of the team by sprint (story points), with which, consequently, it is possible to estimate when a certain functionality that is still in the backlog will be available.

Reduction of risks: The fact of carrying out the most valuable functionalities in the first place and of knowing the speed with which the team advances in the project, allows to clear risks effectively in advance.

Events in Scrum

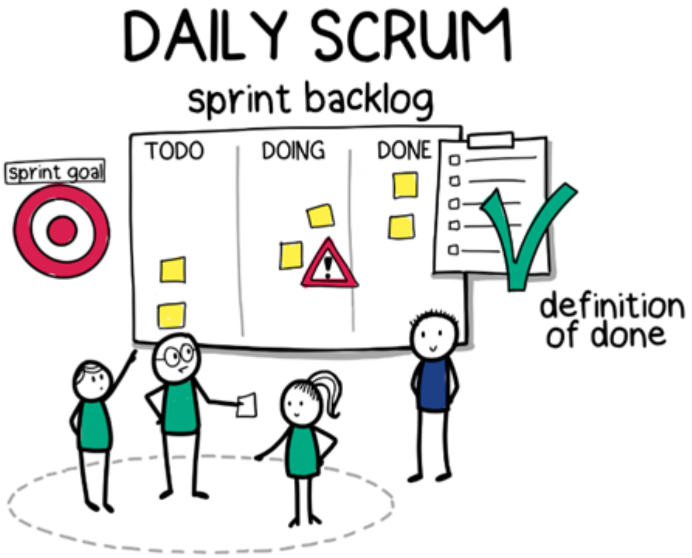

EVENTS/CEREMONIES IN AGILE METHODOLOGY (SCRUM)
1
SPRINT PLANNING
To craft a Sprint Goal and a plan to achieve it
2
DAILY SCRUM
To review team's progress on Sprint Goal and adjust the plan to get there
3
SPRINT REVIEW
To review the work completed in the sprint with stakeholders
4
SPRINT RETROSPECTIVE
To review the product development process and identify areas for continuous improvements
SPRINT BACKLOG
1.REGISTER
2. LOG IN
★★★★★
★★★★★

Each of the Scrum events facilitates the adaptation of some of the aspects of the process, the product, progress or relationships.

Sprint Planning:

The goal of the Sprint Planning is to define what is going to be done in the Sprint and how it is going to be done. This meeting is held at the beginning of each Sprint and is defined how it will approach the project coming from the Product Backlog stages and deadlines. Each Sprint is composed of different features.

Daily Scrum:



The objective of the Daily Scrum is to evaluate the progress and trend until the end of the Sprint, synchronizing the activities and creating a plan for the next 24 hours. It is a brief meeting that takes place daily during the Sprint period. Three questions are answered individually: What did I do yesterday? What am I going to do today? What help do I need? The Scrum Master should try to solve problems or obstacles that arise.

Sprint Review:

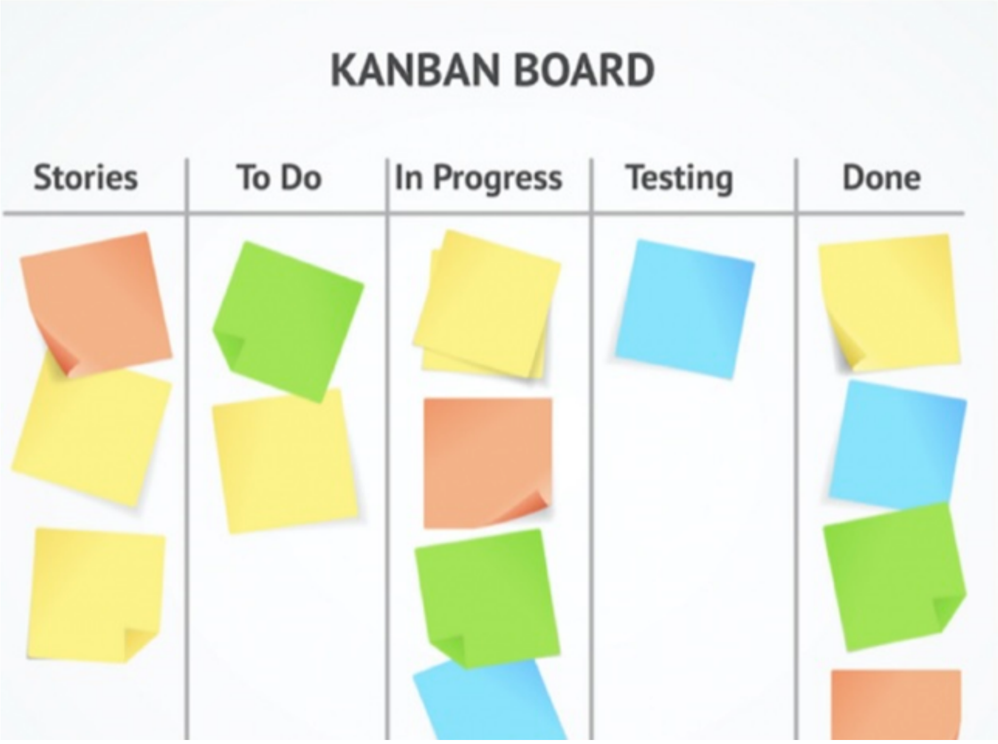
The goal of the sprint review is to show what work has been completed with regards to the product backlog for future deliveries. The finished sprint is reviewed, and there should already be a clear and tangible advancement in the product to present to the client.

Sprint Retrospective:

Good
BAD
?
RETROSPECTIVE
how do YOU think we did?
we were sinking but then learned how to swim!
too many distractions!
coding took a lot longer than usual
PO didn't mess with our plans!
we were a little off with the estimation points...
all bugs were checked in time!

The team reviews the completed goals of the finished sprint, write down the good and the bad, so as not to repeat the mistakes again. This stage serves to implement improvements from the point of view of the development process. The goal of the sprint retrospective is to identify possible process improvements and generate a plan to implement them in the next Sprint.

KANBAN



The Kanban Method is a means to design, manage, and improve flow systems for knowledge work. The method also allows organizations to start with their existing workflow and drive evolutionary change. They can do this by visualizing their flow of work, limit work in progress (WIP) and stop starting and start finishing.

In essence therefore, Kanban is a scheduling system for lean and other Just in Time (JIT) processes. In a Kanban process, there are usually physical (or virtual) “cards” called Kanban that move through the process from start to finish. The aim is to keep a constant flow of Kanban so that as inventory is required at the end of the process, just that much is created at the start.

The core philosophy behind Kanban advocates transparency to the work that is in process through the system so that individuals can interact on the required processes and tools to perform just-enough experimentation instead of settling for status quo! “This lean philosophy is the foundation for the Kanban principles behind the essential Kanban practices of maintaining flow, eliminating waste, and improving continuous learning,” says our Agile Evangelist, Dr. Sriram Rajagopalan.

Kanban Board

Kanban board is used to depict the flow of tasks across the value stream. The Kanban board −

- Provides easy access to everyone involved in the project.
- Facilitates communication as and when necessary.
- Progress of the tasks are visually displayed.
- Bottlenecks are visible as soon as they occur.

Advantages of Kanban board

The major advantages of using a Kanban board are −

- Empowerment of Team − This means −

o Team is allowed to take decisions as and when required.

o Team collaboratively resolves the bottlenecks.

o Team has access to the relevant information.

o Team continually communicates with customer.

- Continuous Delivery − This means −

o Focus on work completion.

o Limited requirements at any point of time.

o Focus on delivering value to the customer.

o Emphasis on whole project.

The tasks and stories are represented by Kanban cards. The current status of each task is known by displaying the cards in separate columns on the board. The columns are labeled as To Do, Doing, and Done. Each task moves from To Do to Doing and then to Done.

Kanban Board is updated on a daily basis as the team progresses through the development.

WIP Limit

The label in the Doing column also contains a number, which represents the maximum number of tasks that can be in that column at any point of time. i.e., the number associated with the Doing column is the WIP (Work-In-Progress) Limit.

Pull Approach

Pull approach is used as and when a task is completed in the Doing column. Another card is pulled from the To Do column.

Self-directing

In Agile Development, the team is responsible for planning, tracking, reporting and communicating in the project. Team is allowed to make decisions and is accountable for the completion of the development and product quality. This is aligned to the characteristic of empowerment of the team in Kanban.

Continuous Flow

In Agile development, there is no gate approach and the work flows across the different functions without wait-time. This contributes in minimizing the cycle time characteristic of Kanban.

Visual Metrics

In Agile Kanban, the metrics are tracked visually using −

- Kanban Board
- Burndown Chart

Uses of Kanban board

Kanban Board is used to −

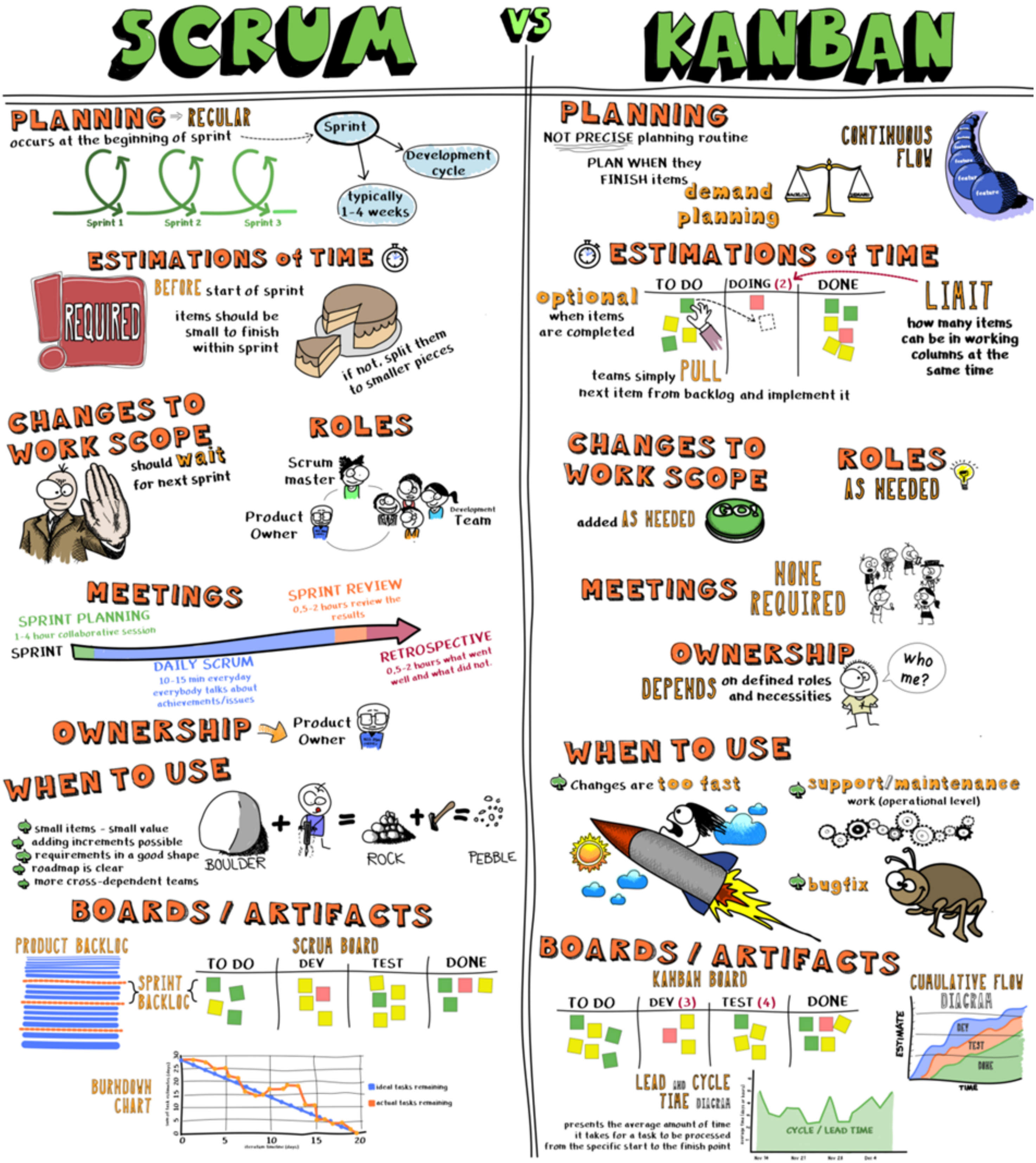
- Measure the cycle times, that can be used to optimize average cycle time.
- Track WIP limit to eliminate waste.
- Track resource utilization to eliminate waste.

Uses of Burndown chart

Burndown chart is used to capture −

- The current status of the tasks and stories.
- The rate of progress of completing the remaining tasks.

As Kanban Board is updated daily, it contains all the information that is required by the Burndown charts.



SCRUM vs KANBAN

SCRUM

PLANNING → REGULAR

occurs at the beginning of sprint → Sprint → Development cycle

typically 1-4 weeks

ESTIMATIONS of TIME

REQUIRED

BEFORE start of sprint

items should be small to finish within sprint

if not, split them to smaller pieces

CHANGES TO WORK SCOPE

should wait for next sprint

ROLES

Scrum master

Product Owner

Development Team

MEETINGS

SPRINT PLANNING
1-4 hour collaborative session

SPRINT

DAILY SCRUM
10-15 min everyday
everybody talks about achievements/issues

SPRINT REVIEW
0,5-2 hours review the results

RETROSPECTIVE
0,5-2 hours what went well and what did not.

OWNERSHIP → Product Owner

WHEN TO USE

- small items - small value
- adding increments possible
- requirements in a good shape
- roadmap is clear
- more cross-dependent teams

BOULDER + = ROCK + = PEBBLE

BOARDS / ARTIFACTS

PRODUCT BACKLOG

SPRINT BACKLOG

SCRUM BOARD

TO DO	DEV	TEST	DONE
🟩 🟩 🟨 🟩	🟨 🟥 🟨 🟨	🟨 🟨 🟩 🟨 🟨 🟨	🟩 🟥 🟨 🟩

BURNDOWN CHART

KANBAN

PLANNING

NOT PRECISE planning routine

PLAN WHEN they FINISH items

demand planning

CONTINUOUS FLOW

ESTIMATIONS of TIME

optional
when items are completed

TO DO	DOING (2)	DONE
🟩 🟨 🟨 🟩	🟥	🟩 🟩 🟨 🟥 🟨 🟨

LIMIT
how many items can be in working columns at the same time

teams simply PULL next item from backlog and implement it

CHANGES TO WORK SCOPE

added AS NEEDED

ROLES AS NEEDED

MEETINGS

NONE REQUIRED

OWNERSHIP

DEPENDS on defined roles and necessities

Who me?

WHEN TO USE

- Changes are too fast
- support/maintenance work (operational level)
- bugfix

BOARDS / ARTIFACTS

KANBAN BOARD

TO DO	DEV (3)	TEST (4)	DONE
🟩 🟨 🟩 🟨 🟨 🟩	🟨 🟥 🟨	🟩 🟨 🟨 🟨	🟩 🟥 🟨 🟩 🟨

CUMULATIVE FLOW DIAGRAM

LEAD AND CYCLE TIME DIAGRAM

presents the average amount of time it takes for a task to be processed from the specific start to the finish point

Previous Page

Company

Techproeducation provides quality online IT courses and coding bootcamps with reasonable prices to prepare individuals for next-generation jobs. from beginners to IT professionals, we offer cutting-edge programs used today by leading corporations.

Contact

☎+1 (917) 768-7466
✆ +1 (917) 768-7466
✉ info@techproeducation.com
New York City, NY USA

Enroll

Enroll Free Course

Enroll Online IT Course

Programs

FREE ONLINE IT COURSES
AUTOMATION ENGINEER
SOFTWARE DEVELOPMENT
CLOUD ENGINEERING & SECURITY
DATA SCIENCE
DIGITAL MARKETING

TECHPROEDUCATION

Reset user tour on this page ℹ