

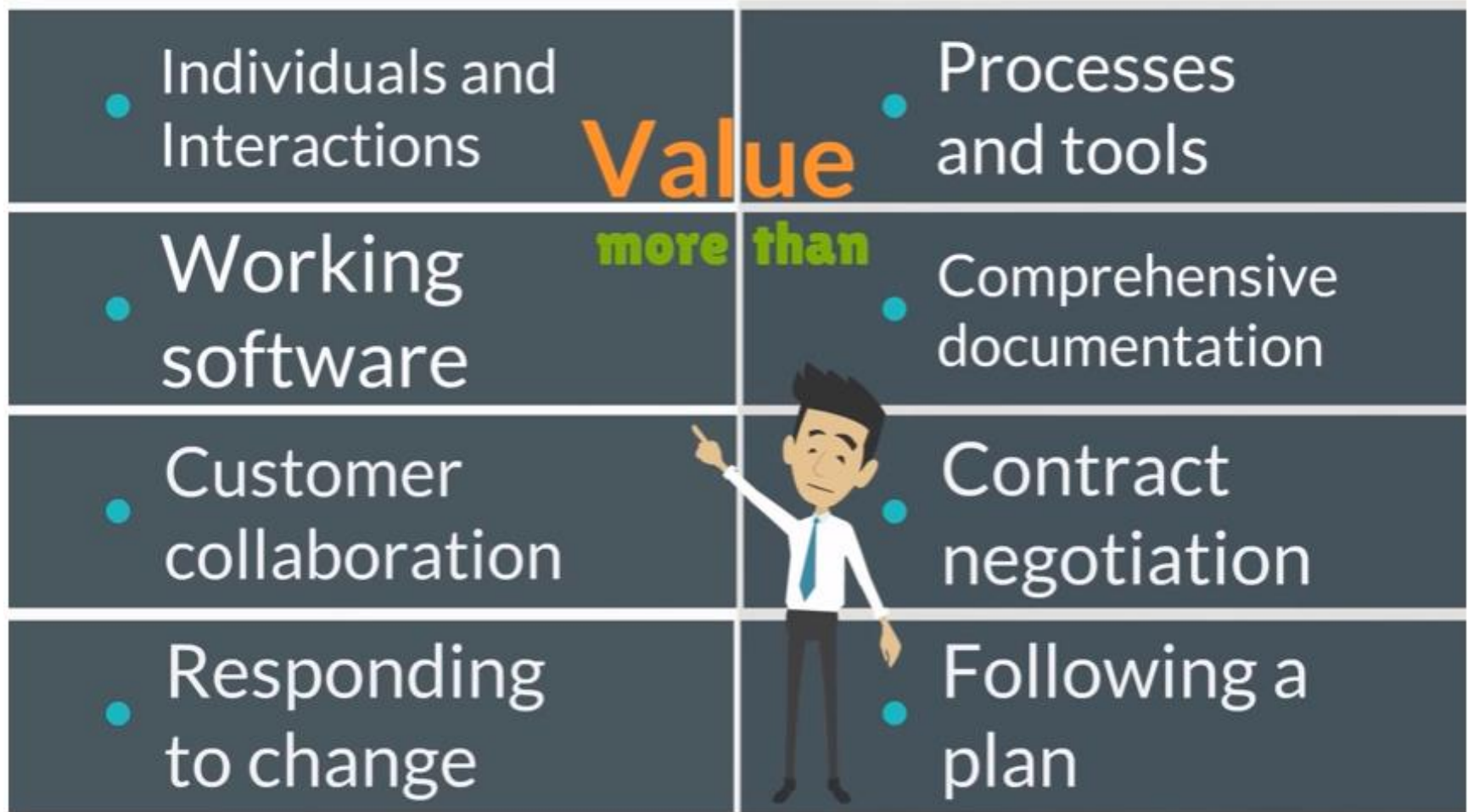
Agile software Testing

Agile Methodology

AGILE methodology is a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project. Both development and testing activities are concurrent unlike the Waterfall model.

- The agile software development emphasizes on four core values.
 - Individual and team interactions over processes and tools
 - Working software over comprehensive documentation
 - Customer collaboration over contract negotiation
 - Responding to change over following a plan

Agile Methodology : Values more for items on left side than right one



What is Agile Testing?

A software testing practice that follows the principles of agile software development is called Agile Testing. Agile is an iterative development methodology, where requirements evolve through collaboration between the customer and self-organizing teams and agile aligns development with customer needs.

Advantages of Agile Testing

- Agile Testing Saves Time and Money
- Less Documentation
- Regular feedback from the end user
- Daily meetings can help to determine the issues well in advance

Principles of Agile Testing

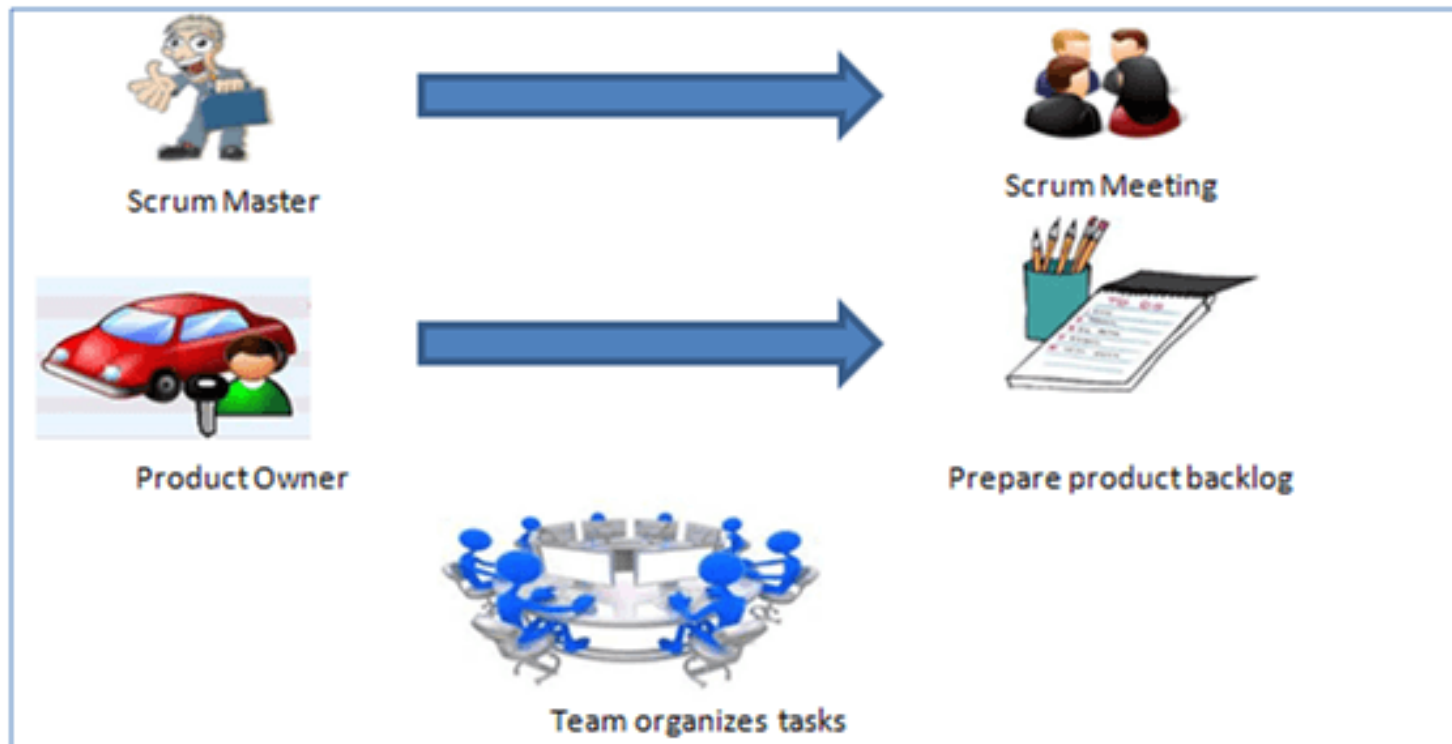
- **Testing is NOT a Phase:** Agile team tests continuously and continuous testing is the only way to ensure continuous progress.
- **Testing Moves the project Forward:** When following conventional methods, testing is considered as quality gate but agile testing provide feedback on an ongoing basis and the product meets the business demands.
- **Everyone Tests:** In conventional SDLC, only test team tests while in agile including developers and BA's test the application.
- **Shortening Feedback Response Time:** In conventional SDLC, only during the acceptance testing, the Business team will get to know the product development, while in agile for each and every iteration, they are involved and continuous feedback shortens the feedback response time and cost involved in fixing is also less.
- **Clean Code:** Raised defects are fixed within the same iteration and thereby keeping the code clean.
- **Reduce Test Documentation:** Instead of very lengthy documentation, agile testers use reusable checklist, focus on the essence of the test rather than the incidental details.
- **Test Driven:** In conventional methods, testing is performed after implementation while in agile testing, testing is done while implementation.
- **More is less : More Interactions and More Focus leads to Less doubts and Less Defects.**

Scrum

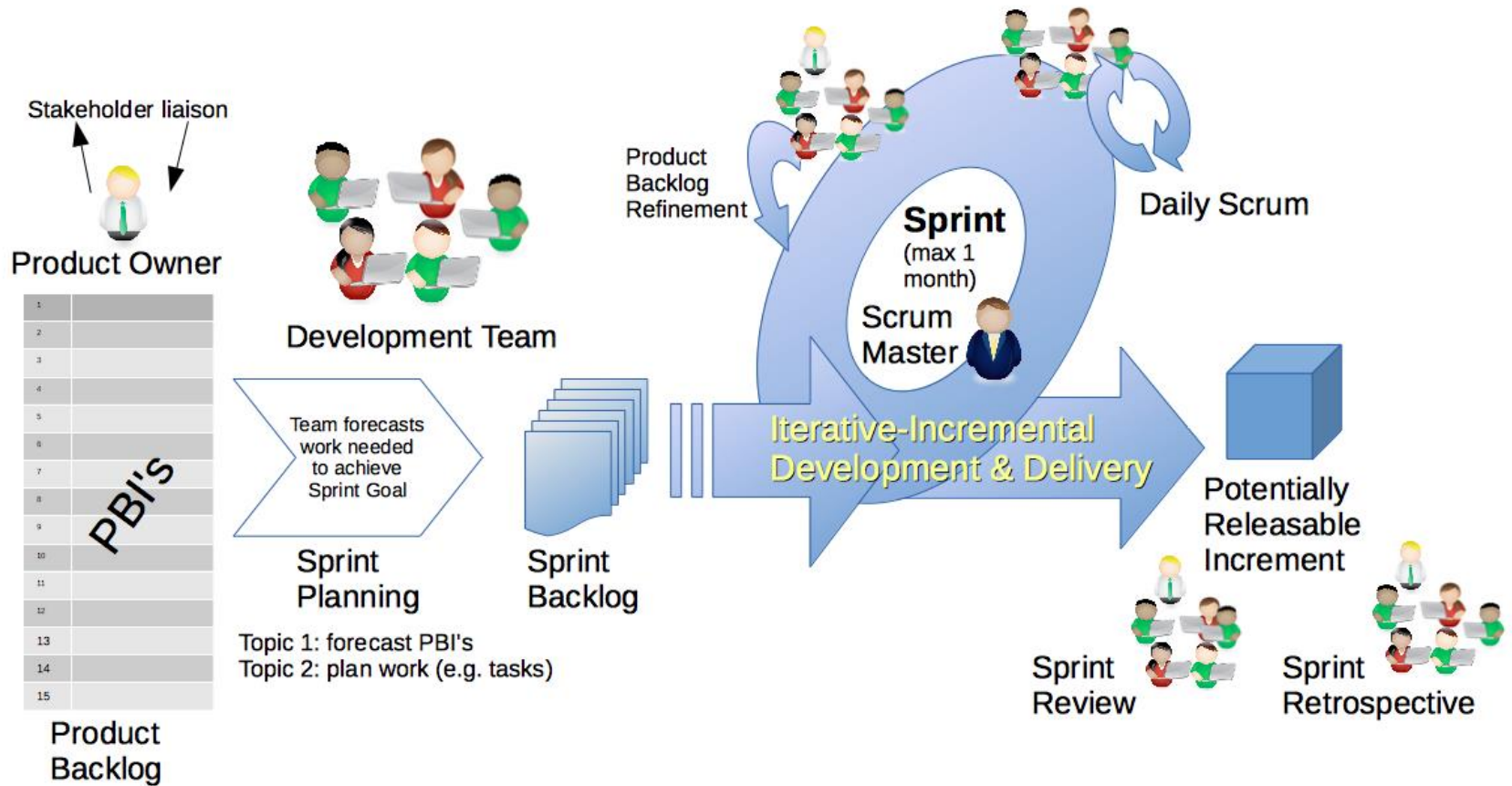
Scrum is a framework for managing software development. It is designed for teams of three to nine developers who: break their work into actions that can be completed within fixed duration cycles (called "sprints"), track progress and re-plan in daily 15-minute stand-up meetings, and collaborate to deliver workable software every sprint.

- SCRUM is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment. Basically, Scrum is derived from activity that occurs during a rugby match. Scrum believes in empowering the development team and advocates working in small teams (say- 7 to 9 members).

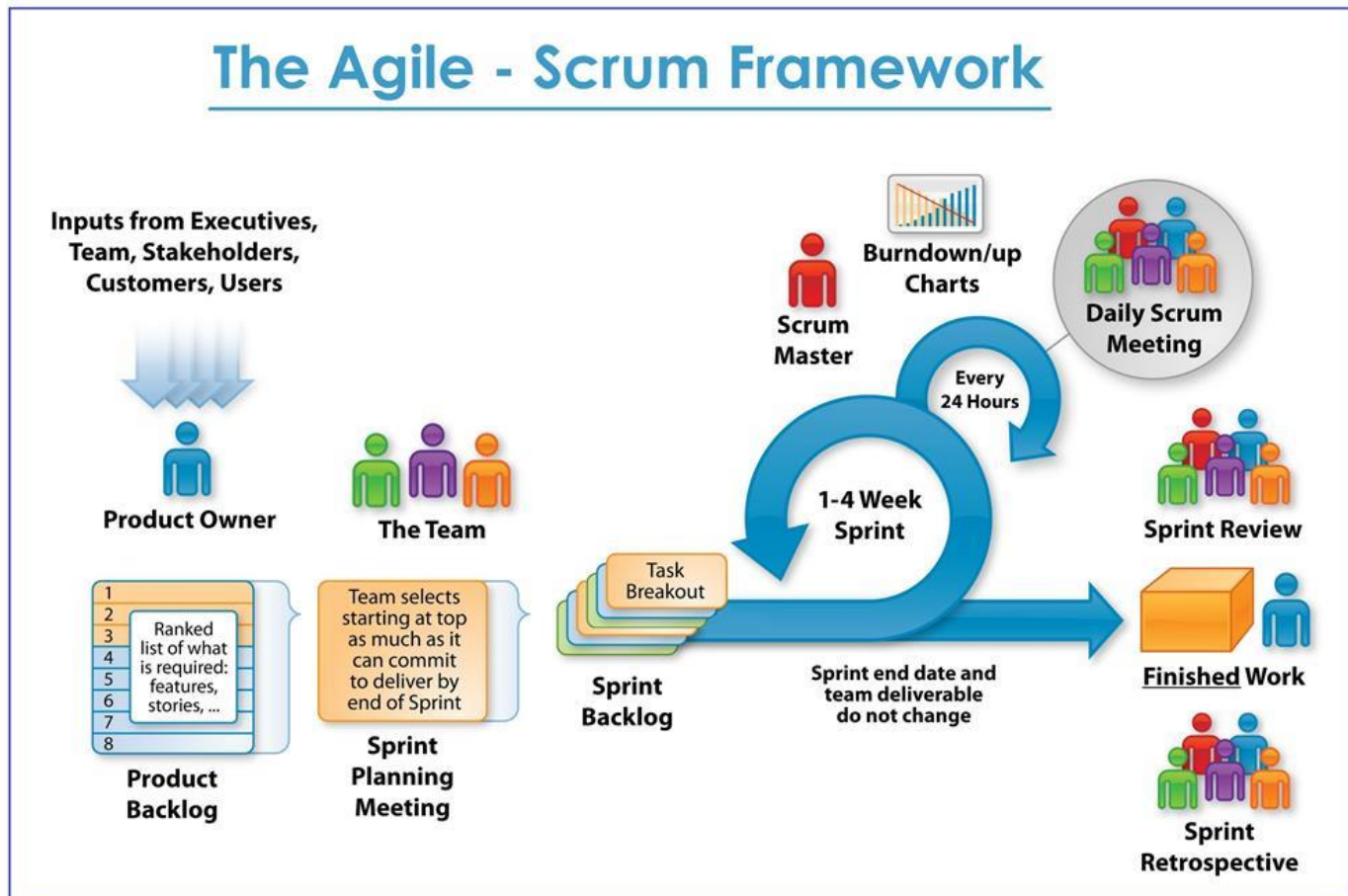
Scrum



Scrum



Scrum

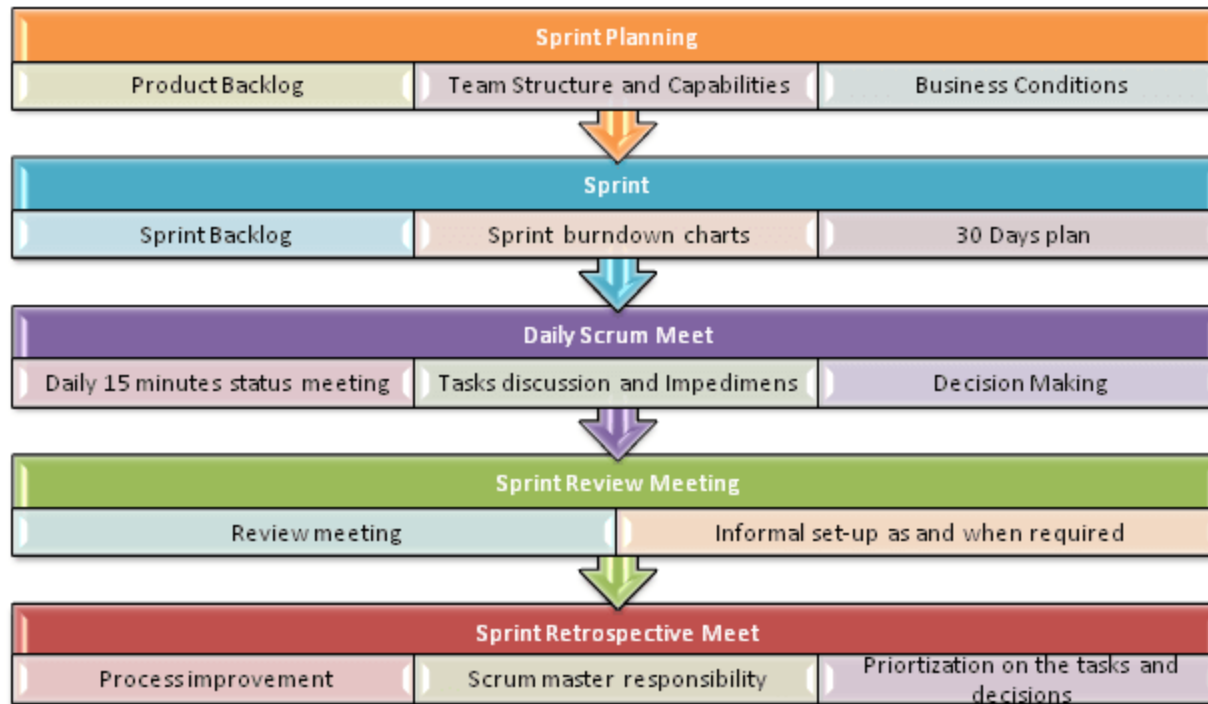


It consists of three roles, and their responsibilities are explained as follows:

- Scrum Master
 - Master is responsible for setting up the team, sprint meeting and removes obstacles to progress
 - Helping the product owner maintain the product backlog in a way that ensures the needed work is well understood so the team can continually make forward progress
 - Helping the team to determine the definition of done for the product, with input from key stakeholders
 - Coaching the team, within the Scrum principles, in order to deliver high-quality features for its product
 - Promoting self-organization within the team
 - Helping the scrum team to avoid or remove impediments to its progress, whether internal or external to the team
 - Facilitating team events to ensure regular progress
 - Educating key stakeholders in the product on Scrum principles
- Product owner
 - The product owner represents the product's stakeholders and the voice of the customer; and is accountable for ensuring that the team delivers value to the business. The product owner defines the product in customer-centric terms (typically user stories), adds them to the product backlog, and prioritizes them based on importance and dependencies. Scrum teams should have one product owner
 - is responsible for the delivery of the functionality at each iteration
- Scrum Team
 - The development team is responsible for delivering potentially shippable product increments every sprint (the sprint goal).
 - The team has from three to nine members who carry out all tasks required to build the product increments (analysis, design, development, testing, technical writing, etc.)

Product Backlog

- This is a repository where requirements are tracked with details on the no. of requirements to be completed for each release. It should be maintained and prioritized by Product Owner, and it should be distributed to the scrum team. Team can also request for a new requirement addition or modification or deletion



- The Sprint Review is equivalent to a user acceptance test.
- Sprint Retrospective is equivalent to a project post-mortem.
- The Sprint Review is focused on the "product" and maximizing the business value of the results of the work of the previous sprint and the Sprint Retrospective is focused on the process and continuous process improvement.

Scrum

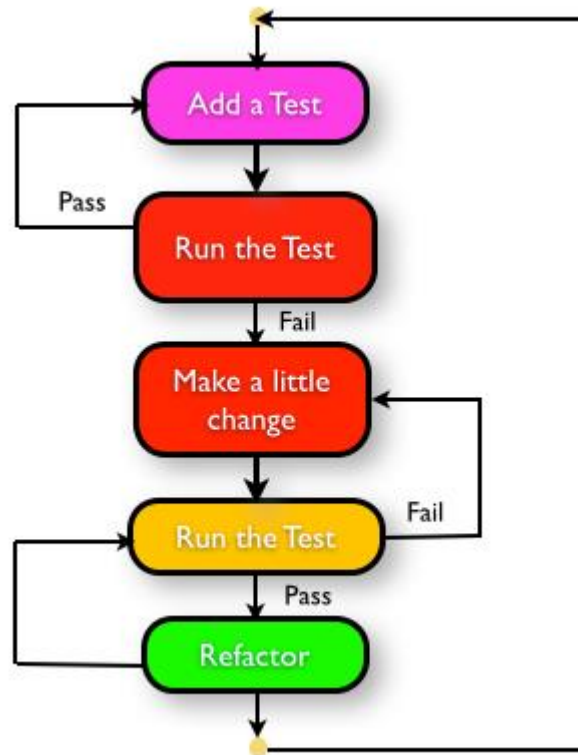
Process flow of Scrum Methodologies:

- Each iteration of a scrum is known as Sprint
- Product backlog is a list where all details are entered to get end product
- During each Sprint, top items of Product backlog are selected and turned into Sprint backlog
- Team works on the defined sprint backlog
- Team checks for the daily work
- At the end of the sprint, team delivers product functionality

Agile Testing: Test-Driven Development (TDD)

- Test-driven development starts with developing test for each one of the features. The test might fail as the tests are developed even before the development. Development team then develops and refactors the code to pass the test.
- Test-driven development is related to the test-first programming evolved as part of extreme programming concepts.
- Test-Driven Development Process:
 - Add a Test
 - Run all tests and see if the new one fails
 - Write some code
 - Run tests and Refactor code
 - Repeat

Test-Driven Development

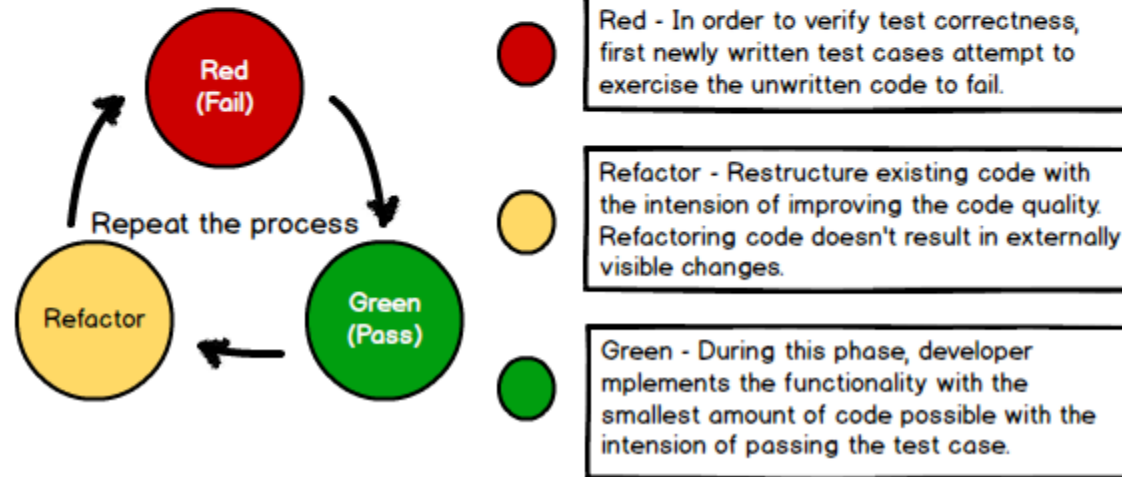


Test-Driven Development

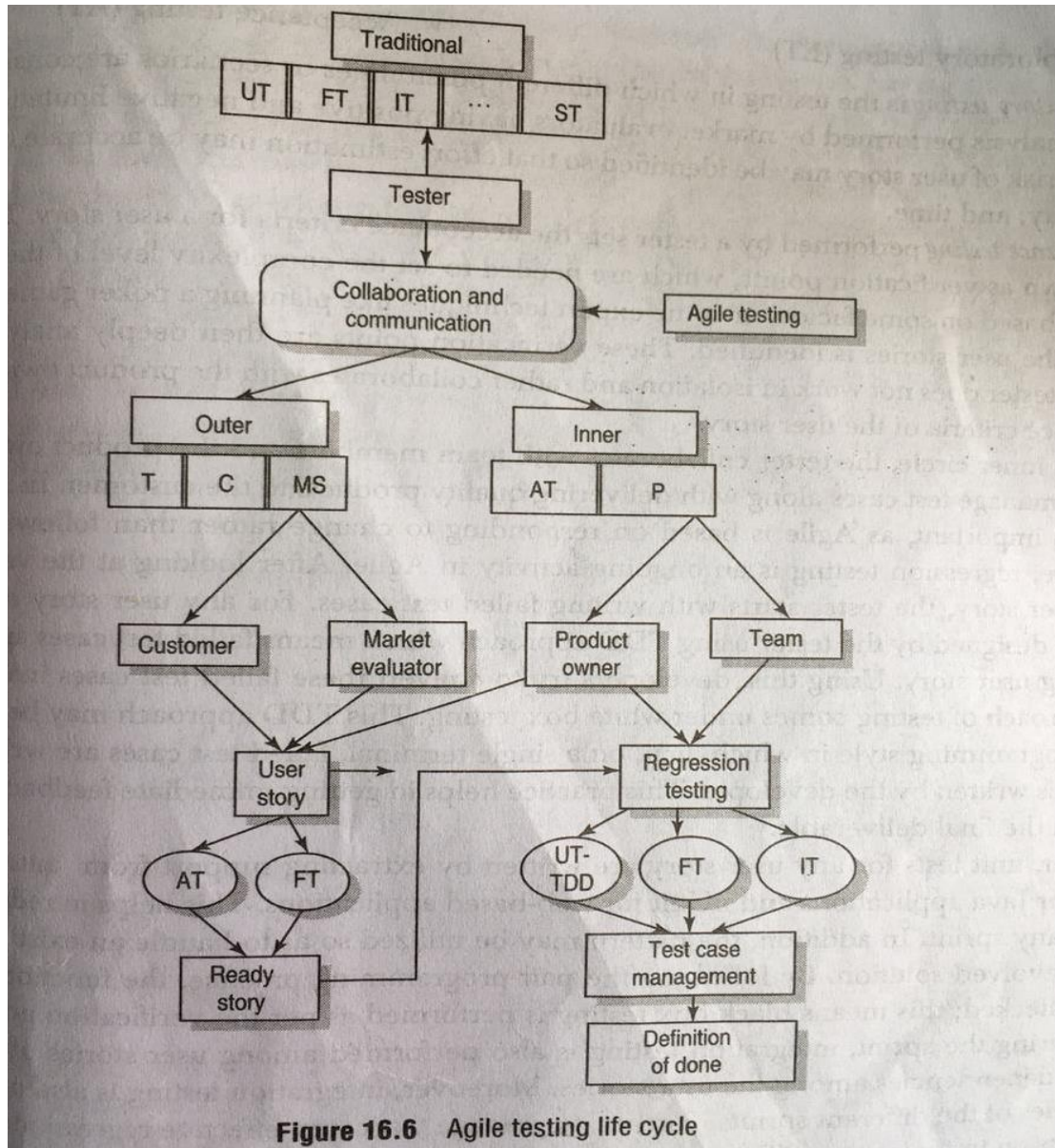
Test Driven Development (TDD)

An Agile Development Approach for

TDD is a quality-focussed, extreme programming practice in which test cases are built prior to feature

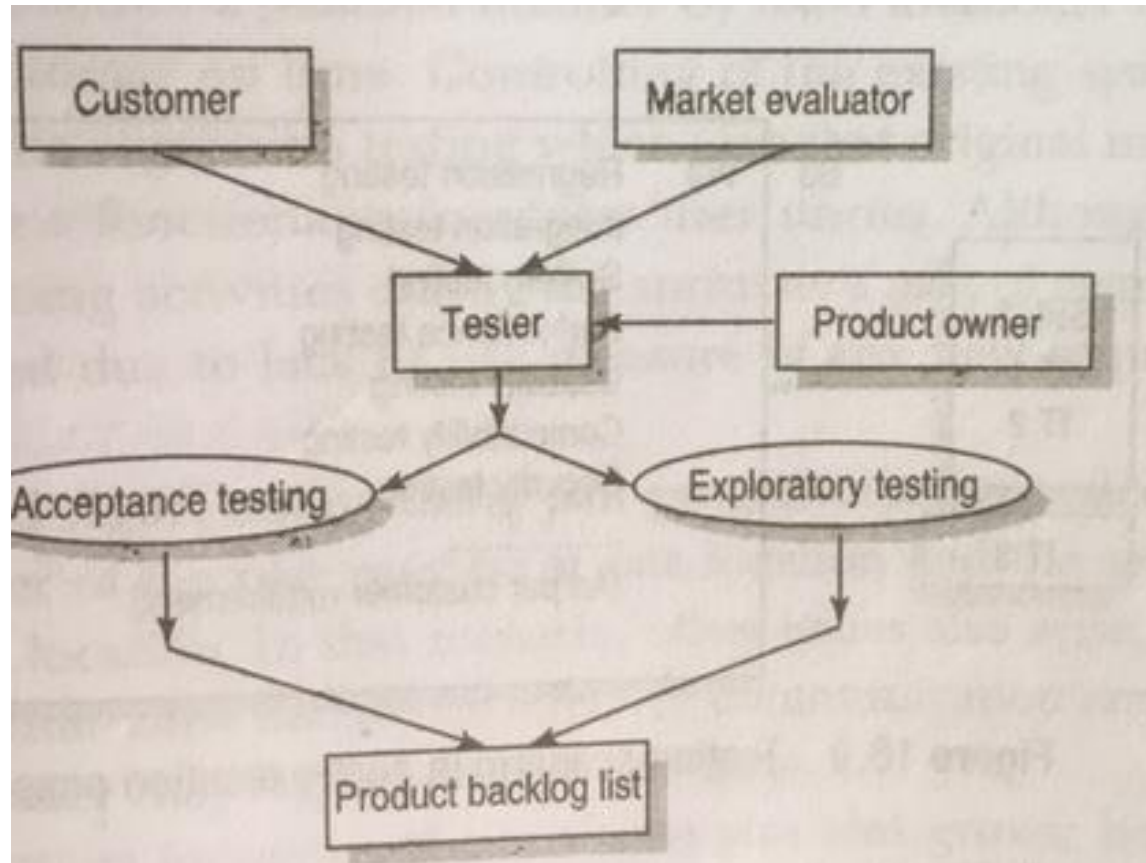


Agile Testing Life Cycle



Reference: Software Testing Principles and Practices, Naresh Chauhan , Oxford University
Pooja Malhotra

Testing in Scrum Phases



Pre- Execution Phase

Reference: Software Testing Principles and Practices, Naresh Chauhan , Oxford University
Pooja Malhotra

Testing in Scrum Phases

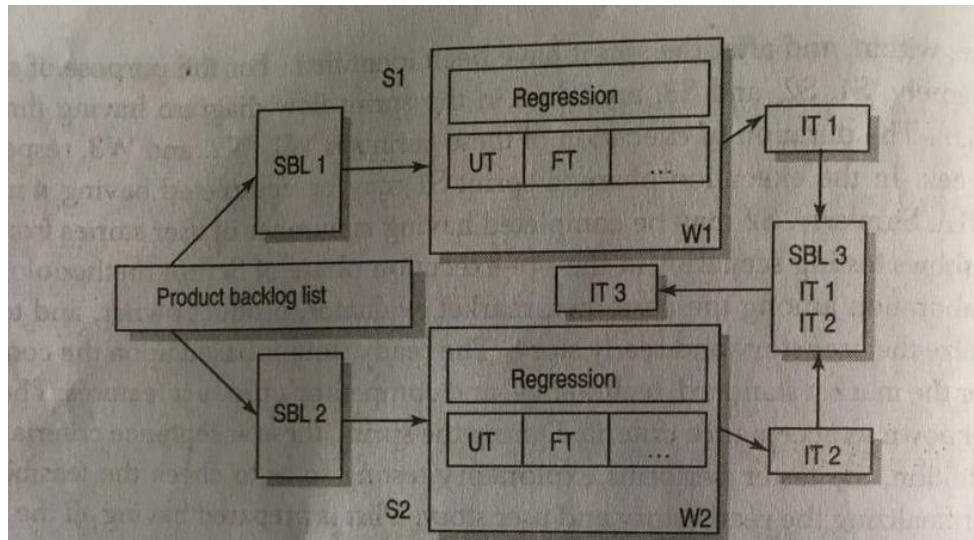


Figure 16.8 Testing scenario in execution phase

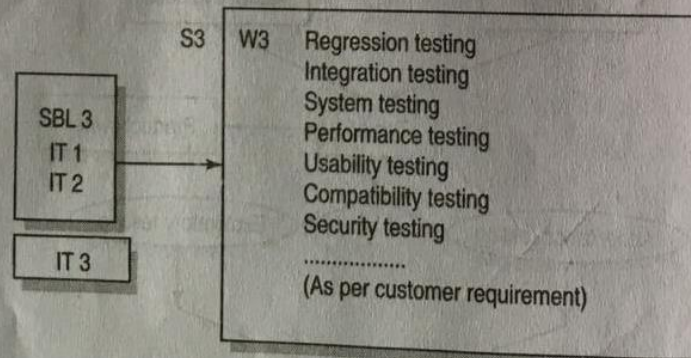


Figure 16.9 Testing scenario in post-execution phase

Execution and post - Execution Phases

References:

Software Testing Principles and Practices, Naresh Chauhan, Second edition, Oxford Higher Education