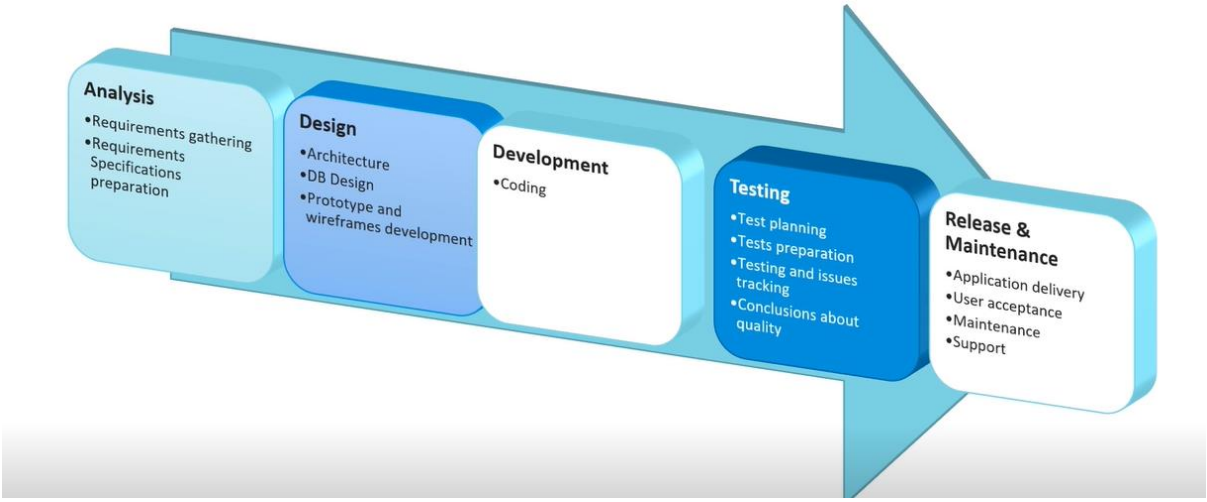
Agile Methodology :

1) Waterfall Model :

Concept



Waterfall summary

FEATURES

- First Process Model
- Each phase must be completed fully before the next phase can begin
- Each phase should be completed with ideal quality
- The whole product is delivered at the end of development process

When to use:

- Requirements are well documented, clear and fixed
- Highly regulated area like government/military/medical...

DISADVANTAGES

- Mistakes may be found later and be very expensive
- Estimates for all product development phases may be inaccurate
- Features are usually more complex than planned
- As results, milestones are often shifted
- Huge amount of documentation
- Year(s) to get version of a product

2) Agile Methodology :

Values

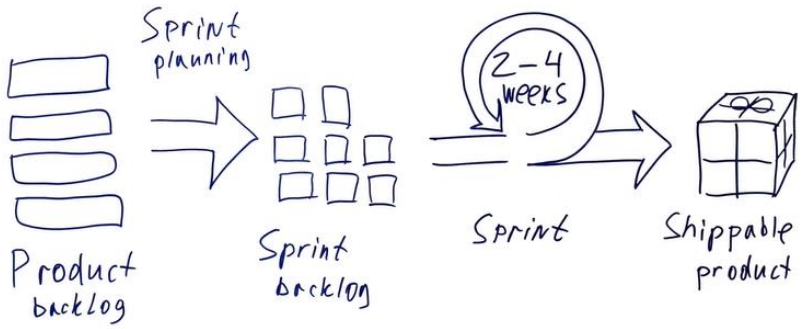
Individuals and interactions	over processes and tool
Working software	over comprehensive documentation
Customer collaboration	over contract negotiation
Responding to change	over following a plan

Principles behind the Agile Manifesto

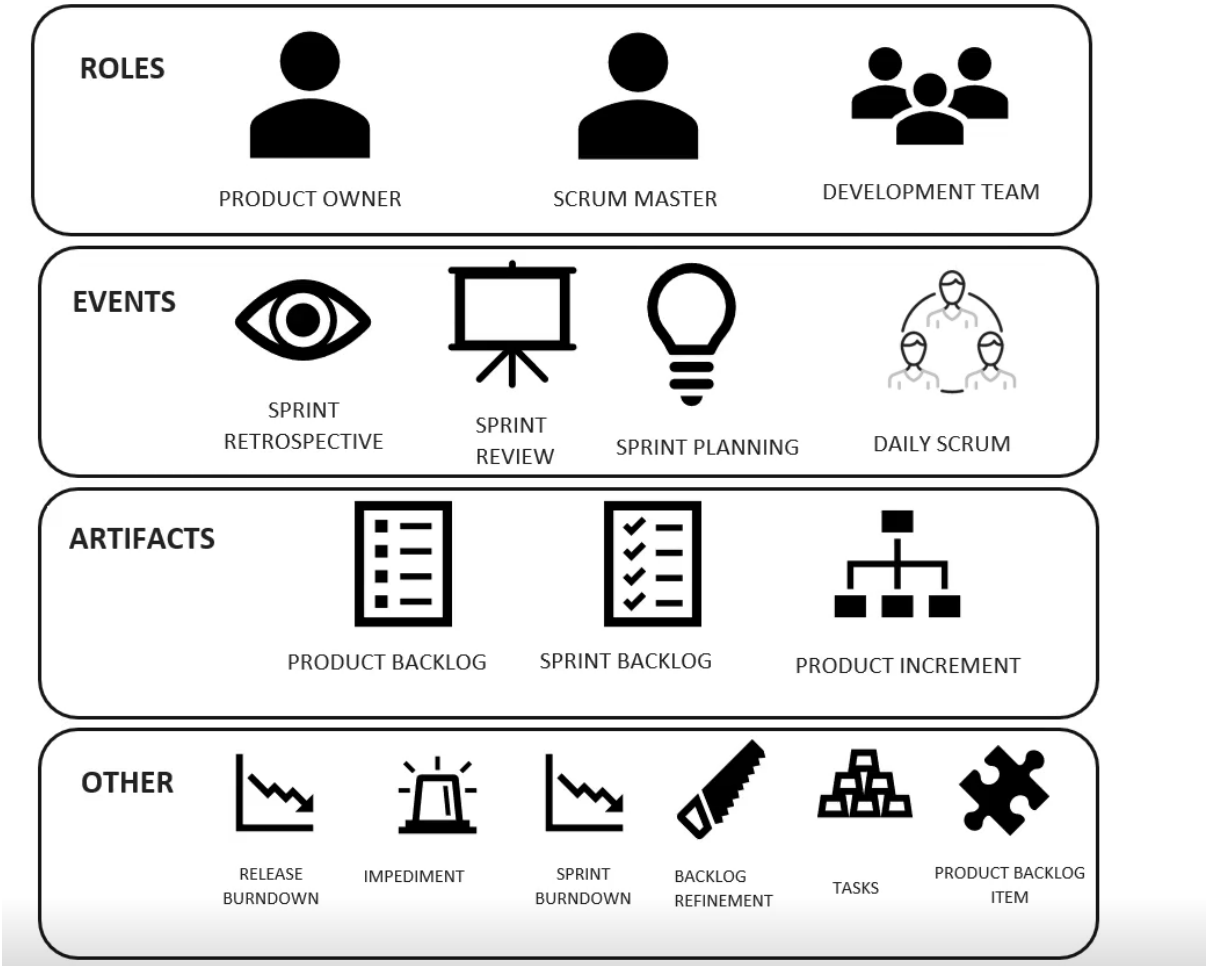
1. The **highest priority** is to **satisfy the customer** through early and continuous delivery of valuable software
2. Welcome **changing** requirements
3. Deliver working software **frequently** (from a couple of weeks to a couple of months)
4. Business people and developers **must work together** 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
8. Agile processes **promote sustainable** development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely
9. Continuous attention to **technical excellence** and **good design** enhances agility
10. **Simplicity** is essential
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.

3) Scrum :

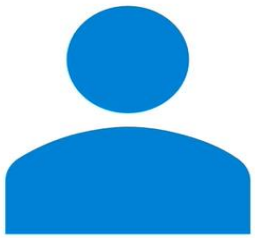
Scrum concept



Scrum elements

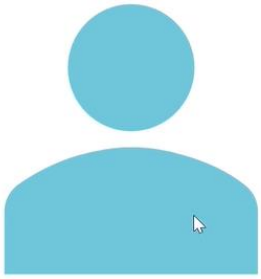


Scrum Roles: Product Owner



- Clears user stories for the development team
- Sets **priorities** for the stories
- Responsible for initial **planning**
- Communicate with the rest of the company

Scrum Roles: Scrum Master



- Responsible for ensuring Scrum is understood and enacted by **Product Owner** and **Development team**
- Resolves team **impediments**

Scrum Roles: Development Team

- Self-organized, small team
- **Cross-functional**, with all of the skills as a team necessary to create a product Increment;
- No titles and sub-teams

Scrum Events: Sprint Grooming (backlog refinement)

Usually performed before sprint

Team:

- clarify requirement to use stories
- ask questions to Product Owner
- define how story can be developed and tested

Scrum Events: Sprint Planning

Usually performed on the 1st day of a sprint

Team:

- reviews high priority stories
- selects stories for sprint
- breaks stories into tasks and estimates tasks execution

Scrum Events: Daily scrum / Standup

- Scrum master tracks and resolves challenges
- 10 – 15 minutes
- Face 2 face communication is appreciated
- Each team member describes:
  - what is done
  - what is planned
  - challenges if any

Scrum Events: Sprint Review / Demo

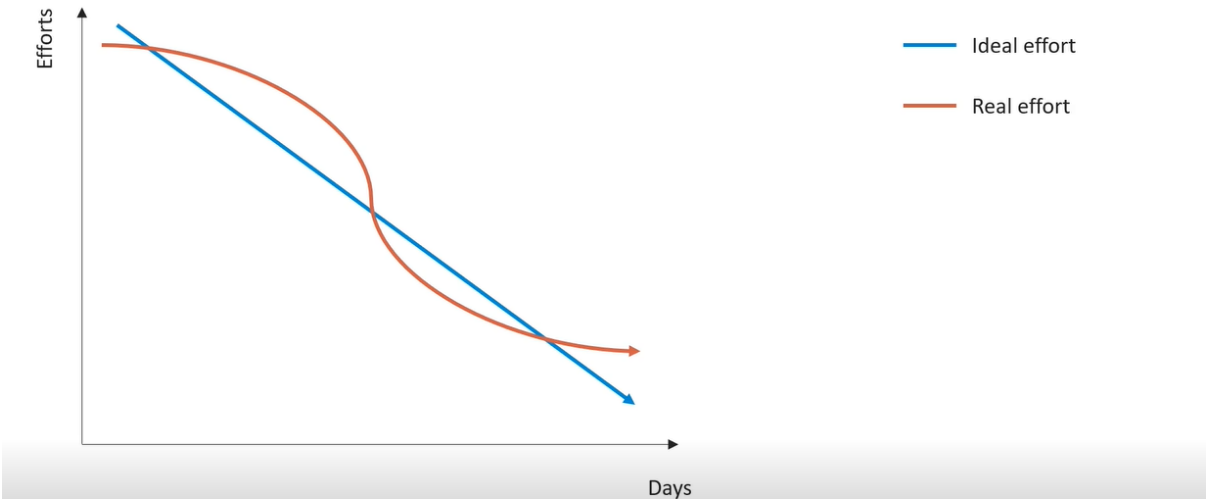
Usually performed at the end of a sprint when scope is ready

- Developer/Tester/BA responsible for the feature **demonstrates** the work that was Done
- Receives **feedback** from customer

Scrum Events: Sprint Retrospective

- What was done well
- What could be done better
- Assign responsible person
- Set timelines for changes

Scrum Other Elements: Burndown chart



What does every team member speak about during stand-up meeting?

1/1 point

- [ ] How did a sprint pass?
- [x] What did he do yesterday?
- [x] What is he going to do?
- [ ] How much time does each story take in a sprint?
- [x] What difficulties does he have?

✔

What does a team discuss during a sprint retrospective?

1/1 point

- [x] What was good about a sprint?
- [x] What was bad about a sprint?
- [ ] How many tasks to take for the next sprint?
- [ ] Which stories will be passed to the next sprint?

✔

How often is stand-up meeting?

1/1 point

- ( ) Twice a day: in the beginning of the day and at the end
- (•) Once a day
- ( ) Once a week, usually on Monday or Friday
- ( ) In the beginning of every sprint
- ( ) At the end of every sprint

✔

What exactly is a perfect burndown?

1/1 point

- ( ) Point
- (•) Right line
- ( ) Parabola
- ( ) Hyperbola

✔

Choose the correct statements

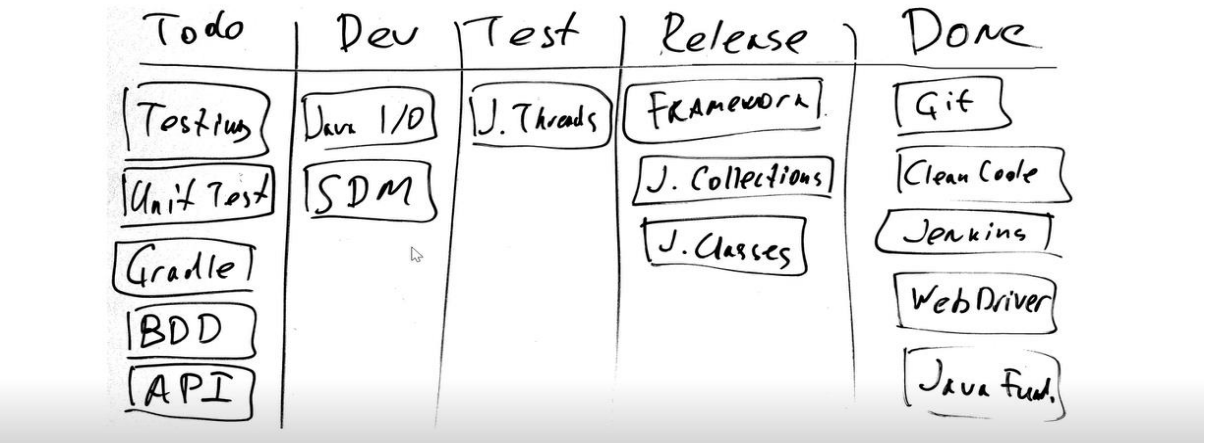
1/1 point

- [x] Product Owner makes the details of the stories clear to the development team
- [ ] Scrum Master decides which stories are going to be carried into the sprint
- [x] Scrum-team usually comprises of about 10 people, more often 4-5 people
- [x] Backlog refinment (grooming) is carried out before the sprint start
- [ ] Stand-up meeting usually takes around an hour
- [x] Sprint Review (Demo) is a meeting during which a team demonstartes the results of its work in a particular sprint

✔

4) Kanban :

Kanban board



Why Kanban

- More flexibility with scope – don't need to fix backlog even for 2 weeks
- Good visualization tasks progress – task board, colors, labels, filters are wildly used
- Easy to start – don't need to set up process according to the framework, move step by steps as you want

KANBAN SUITS FOR THE TEAMS THAT

- Support product
- Have a lot of changes in backlog during sprint
- Do not have clear described user stories in sprint backlog
- Have dependences from other team (separate back-end and front-end dev teams)
- Support other teams like automation/design/DevOps

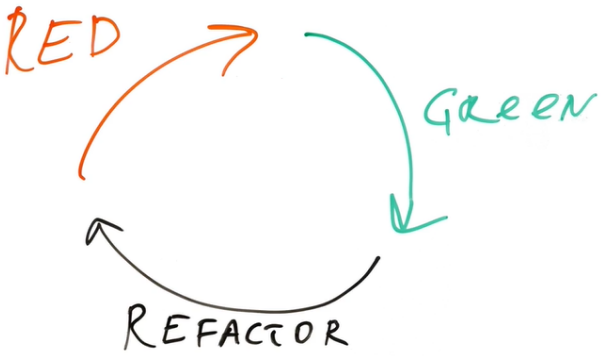
5) Extreme Programming (Xp) :

Extreme Programming

- The Planning Game
- Short Release
- Metaphor
- Simple Design
- Testing: unit tests, automated tests
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40 hours Week
- On-site Customer
- Coding Standards

6) Test-Driven Development (TDD) :

Test Driven Development



- Clean readable well-designed code
- Safe refactoring
- Decrease technical debt on the project
- Stable team velocity
- Many teams report significant reductions in defect rates

- TDD approach requires significant changes from developers and takes time

Select the most right description of the actions within TDD approach.

1/1 point

- ○ Develop a feature then implement testing
- ○ Develop a feature, implement testing, then launch refactoring
- ◉ Implement tests, develop a feature, then launch refactoring
- ○ Launch refactoring, implement tests, develop a feature

✔

How many tests do we need to consider a feature as ready and decently tested?

1/1 point

- One test is enough
- Five tests are enough
- Ten tests are enough
- Depends on the feature ✔

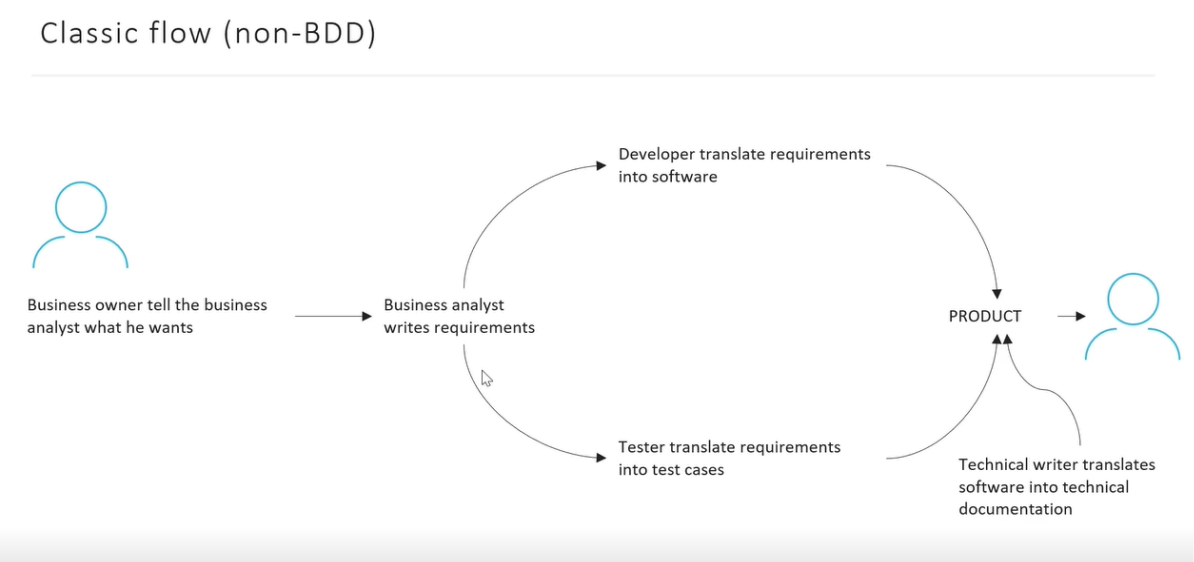
Answer
Correct:
Yes. There are simple features and there are complex ones. There is critical functionality, that can require more thorough testing. TDD practice does not rely on some precise numbers, this is left to developers.

Why are the tests written for functionality development more efficient?

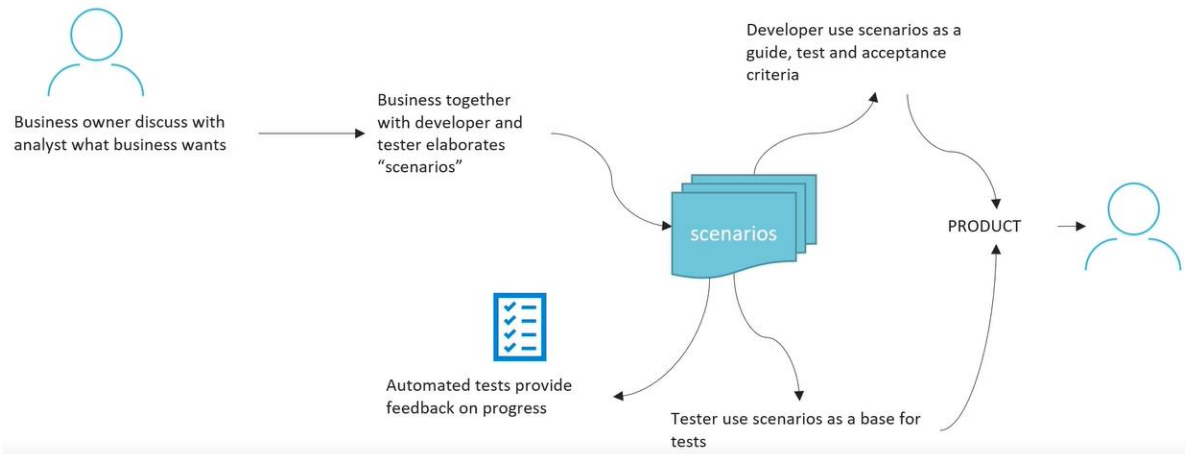
1/1 point

- Developer is not yet tired with coding. The efficiency of a developer decreases in time and by the end of working day it falls to zero or even below zero.
- Developer does not yet know what the implementation will be, and that means that he does not have any prejudices about the algorithm decision implementation. ✔
- Developer always knows what the implementation will be, that's why he implements tests based on the future branches, including those that contain defects.

7) Behavior Driven Development (BDD) :



BDD way



Summary

- Agile is philosophy with values and principals, but not strict instructions how to build process
- Kanban, Scrum are frameworks that have instructions how to set up process
- XP practices, TDD and BDD can be easy integrated into Scrum or Kanban process

Read more