Chapter-7

Date
Page

Software Development Methodologies

→ A software development methodology is a way of managing a software development project.

→ No one methodology is best for all suitation. Even waterfall model (suitable/appropriate for some organization

→ Choosing a suitable methodology can make a big difference in achieving a successful result.

→ ~~As~~

Agile Model: (Move Quickly ~~Quickly~~)

→ Agile methods are those that adapt to changing requirement, minimized development costs, and still give resonable quality software.

→ Agile ~~are~~ project are characterized by many incremental releases each generated in very short ~~period~~ period of time.

→ Advantages of Agile :

(1) Customer satisfaction is rapid, continuous development, and delivery of useful software.

(2) Customer, Developer and Product Owner interact regularly to emphasize rather than processes and tools.

Date
Page

③ Product is developed fast and frequently delivered (weeks rather than months)

④ There is face-to-face communication between clients as face-to-face communication is the best form of communication.

⑤ It continuously give attention to the technical excellence and good design.

⑥ Regular adaption to changing circumstances/requirements.

→

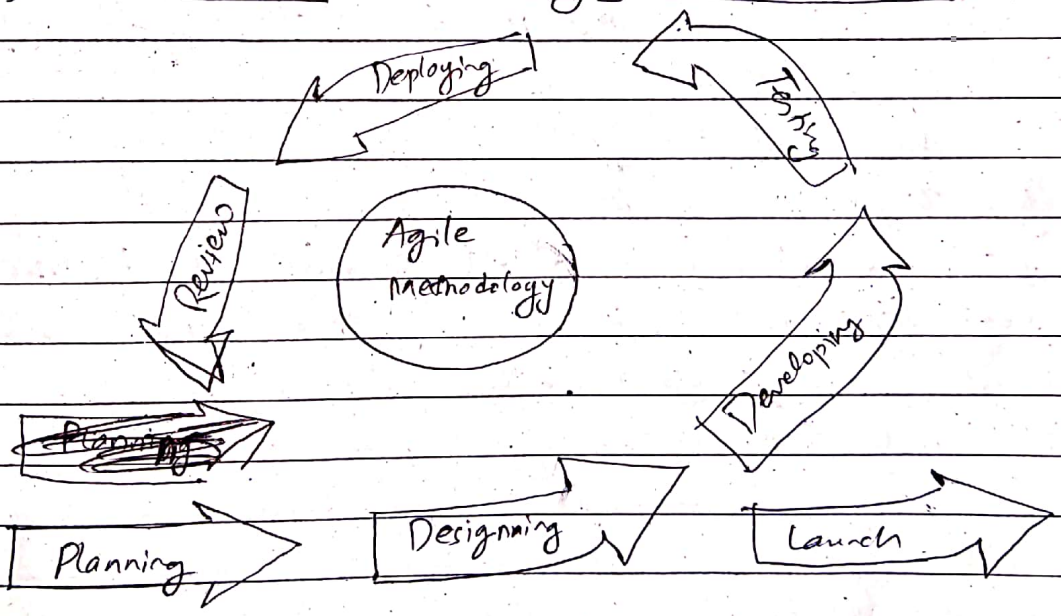


Fig:- Agile Model.

→ Disadvantages of Agile Model:-

① It is not useful for small development projects.

② It is based on assumption that human communication is sufficient for running projects of any size.

③ It requires an expert project member to take crucial decisions in the meetings.

Date
Page

④ Cost of agile Developmat is ~~a~~ methodology is ~~slightly~~ more compared to other methodology.

⑤ They ~~are~~ work poorly ~~with~~ for projects with hundreds of developers or lasting decades.

⑥ The projects can quickly go out off track if ~~man~~ projet manager is not clear about requirements.

→ 1. Key Principles of Agile Methodology are as follows:

① Satisfy customers ~~thro~~ through early and continuous delivery.

② Welcome changing requirements even late in the Project.

③ Deliver value frequently.

④ Build projects around motivated individual.

⑤ The most effective way ~~is of~~ Communication is face-to-face

⑥ Working Software is the Primary Measure of progress.

⑦ Maintain a constant working Pace.

⑧ Simplicity is essential

⑨

* Difference between Agile Method and Waterfall Model :

→ P.T.O

Date
Page

Sn.no	Purpose	Agile Model	Waterfall model
(1)	Definition	Agile model is those that adapt to changing requirements, minimize development costs and still give reasonable quality software.	Waterfall model is one of the easiest and traditional model that has specific deliverables and a review process
(2)	Approach	It follows incremental approach	It follows sequential design process.
(3)	Nature	It is flexible methodology.	It is a structured software development methodology.
(4)	Customer interaction	It has high customers interaction because after every iteration, an incremental version is deployed to the customer.	It has less customers interaction because the product is delivered to the customer after overall development.
(5)	Team size	It has a small team size.	It may consist more team members.
(6)	Suitability	It is not suitable for small projects as expenses is more in agile model.	It is suitable for small projects where requirements is easily understandable
(7)	Test plan	The test plan is reviewed after each sprint.	The test plan is reviewed after complete development.

Date
Page

(Framework)

Different methodologies of Agile Model:-

1. Scrum
2. Extreme Programming (XP)
3. Dynamic System development Method (DSDM)
4. Feature Driven Development (FDD)
5. Rapid Application Development (RAD)

1. Scrum:-

→ Scrum is a subset / framework of Agile.

→ It is a lightweight process framework for agile development.

→ It is most popular implementation of agile ideals.

→ Features are added in short sprint (usually 7-30 days) and frequent short meetings keep people focused.

→ Tasks are generally / usually tracted in scrum board.

→ A scrum process is distinguished from other agile processes by specific concepts and practices, divided into the three categories: Roles, Artifacts and Time Boxes (ceremonies).

Roles :- 3 Roles → Product owner, Scrum Master, Team

Artifacts :- 3 Artifacts → Product Backlog, Sprint Backlog, Burndown Chart.

Time Boxes (Ceremonies) :- 3 ceremonies → Sprint Planning, Daily Scrum, Sprint Review.

discussion

→ Similarity and Difference between Agile & Scrum :-

~~Difference~~ :-

Similarities :-

→ They have some goals and overall objectives.
→ They have collaborative iterations.
→ They focus on creating flexible environment.
→ They both deliver of working software frequently.
→ Both have face-to-face communication with clients.
→ Both develops product fast (usually 7-30 days).

Difference :

	Scrum		Agile
(1)	Scrum methods will always be Agile.	(1)	Agile methods may not always be Scrum.
(2)	~~Is~~ Scrum encourages self-organizing, cross-functional unit.	(2)	Leadership plays a crucial part in Agile process.
(3)	Provides ways to manage project efficiently to meet project goals.	(3)	Doesnot proved any specific set of rules to follow to manage the project.
(4)	Scrum is subset of Agile	(4)	Agile is superset of Scrum.

Date
Page

(2) Extreme Programming (XP):-

→ ~~It framework of~~

→ It is framework of agile. It is a frequent release development methodology in which developers work in pairs for continuous code review.

→ Advantage:-

- Robust, high quality software

→ Disadvantage:-

- Expense is twice the development cost.

→ Most is emphasize on test-Driven Development (TDD).

(3) Dynamic System Development Model:- (DSDM)

→ It is an agile method that sets time, quality, and cost at the beginning of the project.

(4) Feature Driven Development (FDD):

→ It is an iterative Development process with more emphasis on planning out the overall architecture, followed by implementing features in logical order.

importance

(5) ~~RAD~~ Rapid Application Development (RAD):-

→ It is minimalist agile method that emphasis on minimizing planning, a focus on prototyping and using reusable components.

Test Driven development : (TDD)

→ TDD is an iterative development process in which every iterations starts with a set of tests for a new piece of functionality.

→ Test cases are created before code is written.

→ TDD instructs developers to write new code if the test has failed. automated

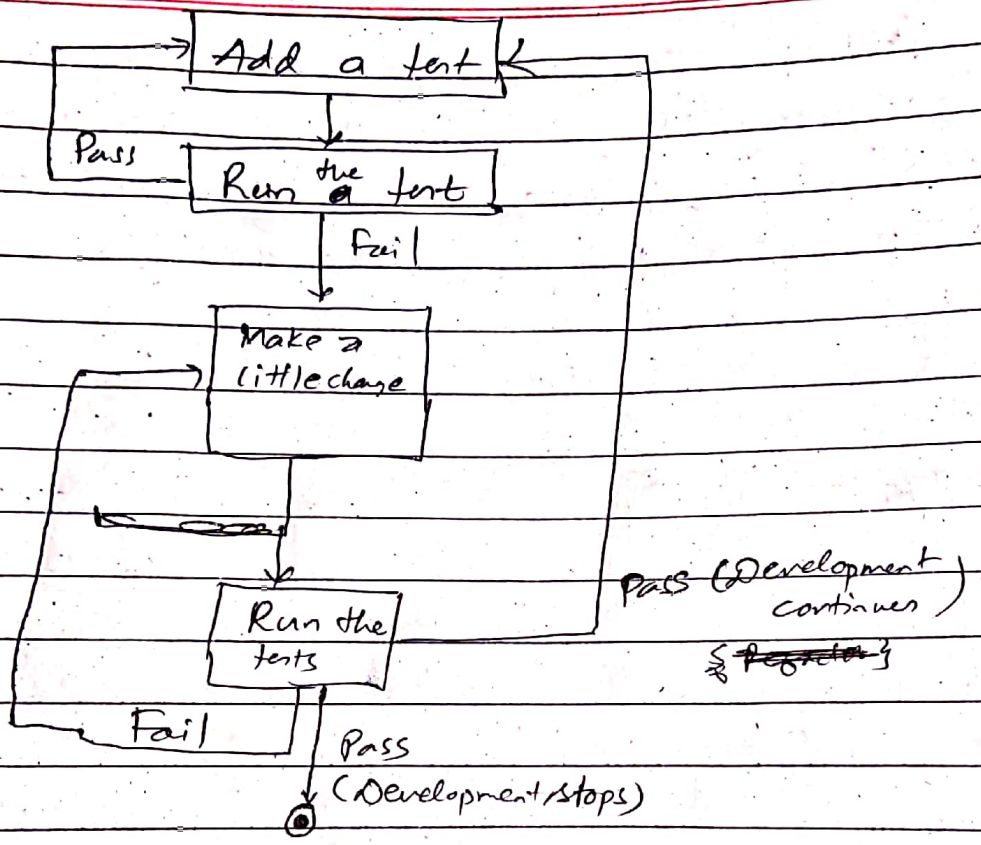
→ Advantages of TDD :

1. Avoids duplication of code
2. Since we are doing Test first there is reduction of bugs.
3. TDD helps to ensure quality by focusing on requirements before writing the code.
4. TDD keeps code clear, simple and testable by breaking it down into small achievable steps.
5. TDD provides documentation for different team members.
6. TDD provides repeatable regression tests.

→ Disadvantages of TDD :

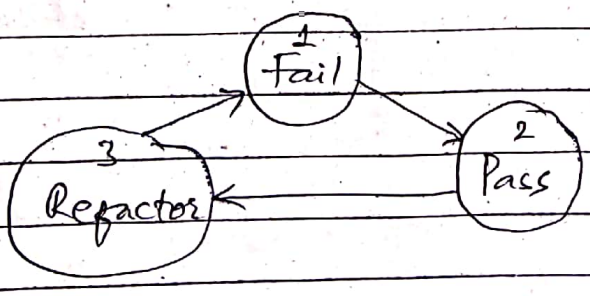
1. Early test cases are heavy in maintenance.
2. Requirement change is paid in TDD.
   If the requirement changes, then test case changes, maintence will change, development effort will change.

→ The steps of Test-first development (TFD) or TDD. :



→ TDD can be described using following formula :

TDD = Refactoring + TFD



→ How to perform TDD Test ?

Add Test

↓

Execute

↓

write code

↓

Repeat

Date
Page

→ There are two level of TDD :-

ⓐ Acceptance TDD (ATDD) :

ⓑ Developer TDD

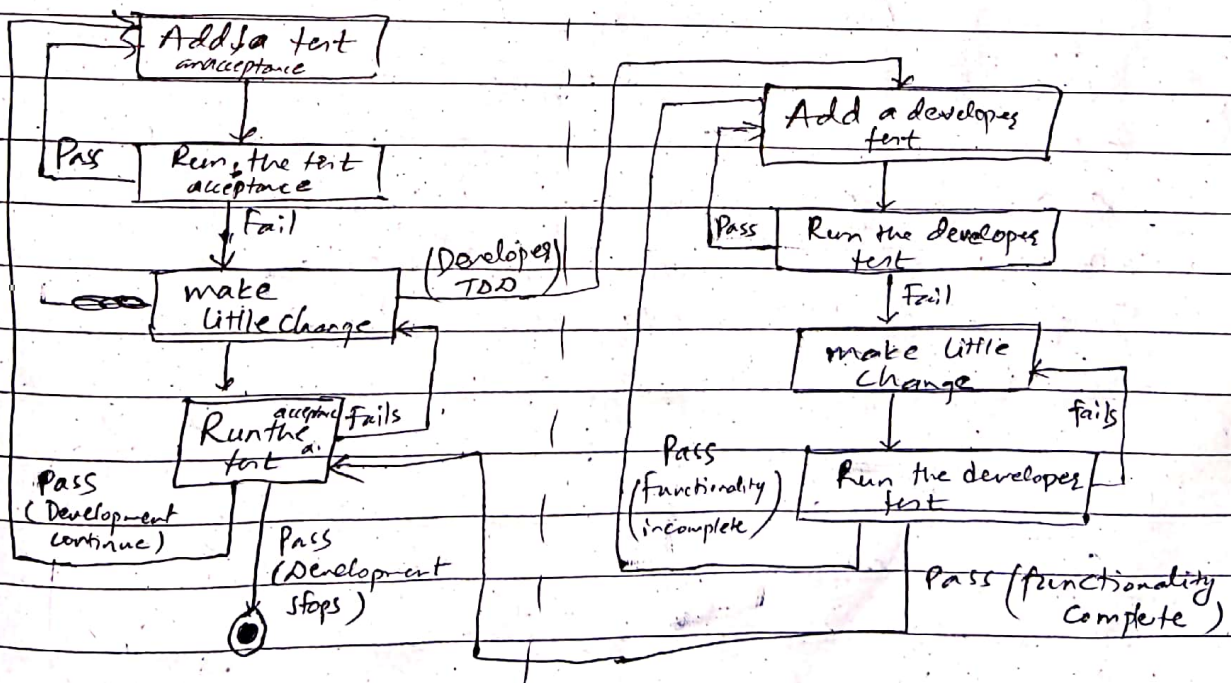
ⓐ Acceptance TDD :- (black-box)

→ With ATDD we write single acceptance test, or behavioral specification depending on your preffered terminology ~~for~~ ' } Source code kaina ~~tested~~ front ~~the~~ layer ko matra test }

ⓑ Developer TDD :-

→ With DTDD we write single developer test, sometimes refered as unit test.

How Acceptance TDD and Developer TDD works

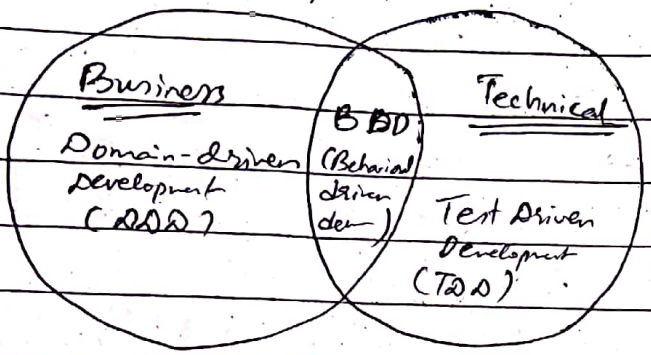


Acceptance TDD

Developer TDD.

Behavior Driven Development - (BDD)

→ BDD is a way for software teams to work that closes the gap between business people and technical people.



→ Example:

Feature : Login functionality

As a costumer ~~I~~ in order to use the application I want to login with my ~~account~~ email & password.

Scenario : Login with invalid credentials.

Given I on at account/login page

When I fill account email textbox with value "incorrect@gmail.com"

AND I fill password ~~of~~ textbox with value "incorrect password".

~~AND~~ AND I click the "login" button.

Then a text " Can't login ! Wrong email or password" is should appear in the validation ~~essess~~ region.

Scenario : Login with valid credentials.

Given I am at account/login page

When I fill account email textbox with value " valid@gmail.com".

And I fill password text box with value "valid password"

AND I click "login" button

Then I should be on the home page.

→ Benefits of BDD :-

(a) ~~Strong Collaboration~~ :

(a) Encouraging collaboration across the roles to build shared understanding of the problem to be solved.

(b) Producing system documentation that is automatically checked against the system's ~~from~~ behavior.

(c) Working is rapid.

→ Disadvantage :-

(a) To ~~working~~ work with BDD, prior exposure to TDD.

(b) BDD is not ~~applicable to for~~ compatible with waterfall approach.

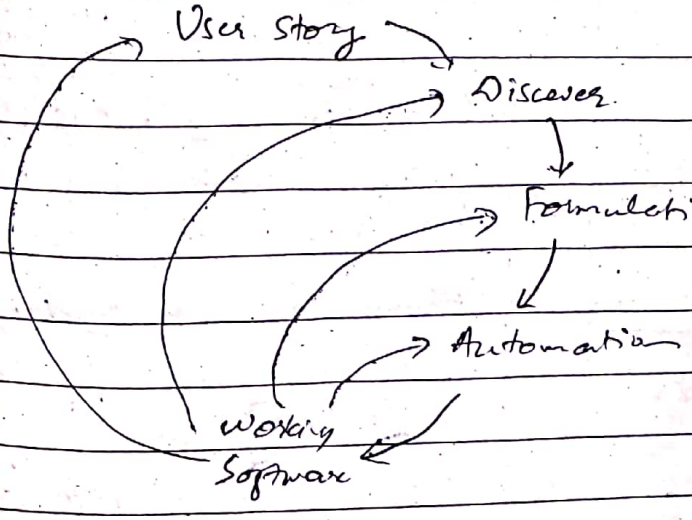
(c) Testers using BDD must have ~~to~~ sufficient technical expertise.

→ Implementing BDD approach :-

(a) Discovery - What it could do.

(b) Formulation - What it should do.

(c) Automation - What it actually does.



Date
Page

difference with ATDD ÷

- focuses more on capturing the requirement
- All stakeholders participations
- Simple English
- ~~use~~ focus on writing acceptance.

The Differences ÷

(1) TDD & BDD

TDD	BDD
(1) TDD stands for Test driven development	(1) BDD stands for Behavioural driven development.
(1) The ~~Test~~ processes starts by writing test case.	(1) The process starting by writing behaviour scenario as per the expected.
(2) TDD focuses on the implementation of a feature.	(2) BDD focuses on the system's behavior.
(3) Test cases are less readable by non-Technical person as they are written in a Programming language.	(3) Scenarios are more readable than TDD as they are written in simple english format.
(4) Collaborations is only required between developers.	(4) Collaborations is required between all the stakeholders.
(5) TDD reduces bugs in our tests.	(5) Bugs in tests are difficult to track when compared to TDD.
(6) The main focus on TDD is unit testing.	(6) The main focus on BDD is understanding requirements.
(7) Some ~~test~~ tools which supports TDD are ÷ JUnit, NUnit, etc. TestNG, etc.	(7) Some tools which supports BDD are, cucumber, Specflow, NSpec, etc.

Date
Page

(2) TDD Vs ~~Test dd~~ Traditional Testing :-

	TDD		Traditional Testing
(1)	In TDD, complete coverage of the testing is achieved. Unlike Tradition testing, in TDD we ~~test~~ test each line of code.	(1)	Focuses on code correctness, but may not detect all coding defects.
(2)	It is cyclical process. (Test - Code - refactor)	(2)	It is linear process. (Design - code - Test)
(3)	It is test - (~~test~~ first approach. (Test cases and testing are done before writing code.)	(3)	It is test - last approach. (first developers write code and the testing is performed at the end of development process)
(4)	It provides cleaner code as there is less bugs.	(4)	It provides less clean code as it may have more bugs compared to TDD

(3) TDD Vs AMDD / ADD (Agile method / Agile driven development)

	TDD		Agile driven Development
(1)	It shrinks programming ~~feedb~~ feedback loops.	(1)	It shrinks modelling feedback loop.
(2)	It is oriented non-visually	(2)	It is oriented as visually.
(3)	It directly communicates with programmer.	(3)	It communicate with stakeholders, business analysts, & data professionals.
(4)	Its scope is limited to some software works.	(4)	Its scope is wider than the TDD & includes stakeholders.

Date
Page

give importance.

(5)	It emphasizes the development of high-quality code.	(5)	It emphasizes quality communication with between stakeholders & developers
(6)	It has detailed specifications	(6)	It emphasizes bigger issues