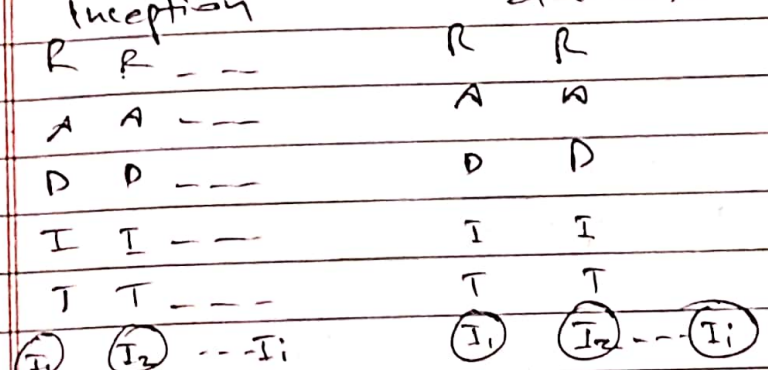
Chapter 4

Software Development Model (SPLC)



4 phases in SPLC (Inception, Elaboration, Construction Transition).

6 processes/phases within each above phase (Requirement gathering / Analysis, Design, Implementation, Testing Deployment / Maintainance).

various methodologies or S/W development model like:

1. waterfall model
2. V model (Verification/validation)
3. Incremental model.
4. RAD model
5. Agile model
6. Iterative model
7. Spiral model
8. Prototype model

① Waterfall Model:

Requirement gathering and Analysis
↳ System design
↳ Implementation
↳ Testing
↳ Deployment
↳ Maintainance

=> generated for small project.
-> testing start after completion of project.
-> phase shouldn't be overlap.
-> Each phase start only after completion of previous model.

② V-model:

- verification and validation
- Sequential Development model

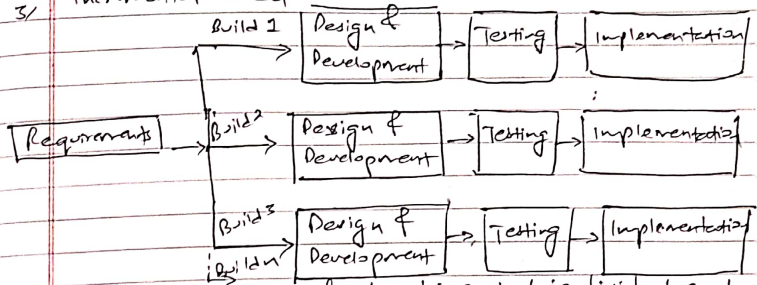
Developers life cycle (verification phase)	Testing life cycle (validation phase)
Business req. specification	→ Acceptance testing
SRS (System Requirement Specification)	→ System testing
high level design	→ System integration testing
low level design	→ Component testing
coding	→ Unit testing

code

fig: V-model.

=> phase shouldn't be overlap
=> proactive defect tracking
=> testing should be in parallel
=> suitable for small project.

③ Incremented model



=> also called multiwaterfall => big project is divided and
=> more flexible, less costly, test & debug separately to fulfill req
=> should be aware on integration.

④ Iterative model

Req. gathering
↓
Analysis
↓
Design 0
↓
Implementation
↓
testing 0
↓
Analysis 0

Design 1 → Implementation 1 → Testing 1 → Analysis 1 - - -

Design n → Implementation n → Testing n → Analysis n

-> development begins as just part of software
-> supports overlapping of phases.

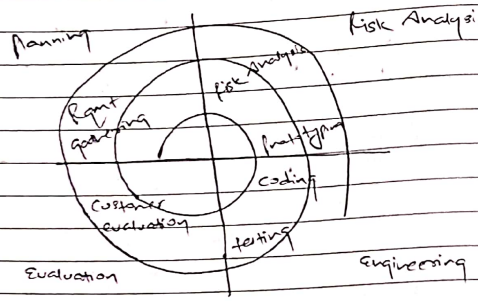
⑤ RAD model (Rapid Application Development):

Team 1	Team 2	Team 3
Business modeling	B.M	B.M
↓	↓	↓
Data modeling	D.M	D.M
↓	↓	↓
process modeling	P.M	P.M
↓	↓	↓
Application generation	A.G	A.G
↓	↓	↓
testing and turnover	TT	TT

<- 60 or 90 days ->

-> It has reduced development time
-> Rapid development

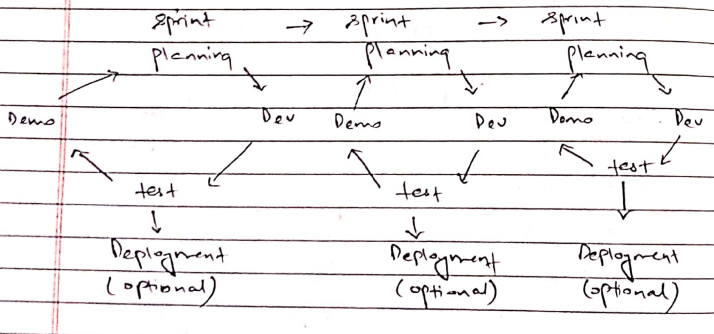
* Spiral Model



* incremental growth in iteration
* more emphasis on "risk analysis"
* better for large project.
* S/w is produced in early phase.

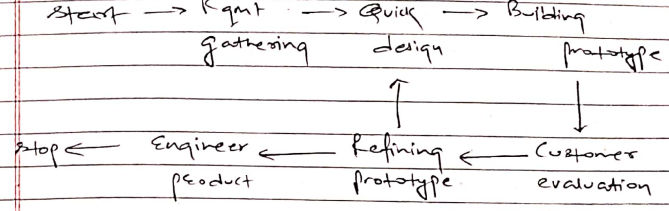
* Agile Model

Kick off



- incremental process with each release building on previous functionality.
- each release is thoroughly tested.
- more emphasis on interaction.

* Prototype Model:



- basic idea is instead of freezing the requirements before design can proceed, a prototype is built to understand the rqmts
- goal is to provide a sys. with overall functionality.

* Software testing Levels:

1) Unit testing:
   - white box testing
   - class/ function method
   - developer
   - smallest unit of s/w

2) component testing
   - a.k.a module testing.
   - after unit testing, it comes into picture
   - done by tester.

- finds defects in module.
- may be done in isolation.

3. Integration testing
- integrates & assesses(check) interfaces and functionality between components
- done by test team.
- different approaches can be used for testing.
- the greater the scope of integration, the more difficult it becomes to isolate failures.

4. System testing
- Concerned with whole system's behaviour
- mostly checked the sys. to verify it meets the specified specifications.
- can use various approaches of testing

5. Acceptance testing
- done by user/customer
- to establish confidence in the system
- most focused on validation type testing.
- types:
  - * user acceptance test
  - * operational " "
  - * contract " "
  - * compliance " "

* Software Testing Types:
- focused on particular test objective
- types:
  - * functional testing
  - * non-functional testing
  - * structural "
  - * change-related "

* functional testing
- activities that verify a specific action or function of code
  - 'does this particular feature work?'
  - can be done as
    - ⊗ regmt based testing
    - ⊗ business process based testing

* non-functional testing:
- may not be related to a specific function/action
- types:
  - → * functionality → efficiency → baseline
  - → reliability → maintainability → compliance
  - → usability → portability → documentation
  - → endurance → load → performance
  - → compatibility → security → scalability
  - → volume → stress → recovery
  - → internationalization & localization.

* structural testing:
- a.k.a white box / clear box / glass box
- testing of internal structure
- testers need to have knowledge of internal

implementation of code.

* change related Testing
  ① Retesting
  ② Regression testing.

Chapter: 4
Testing throughout the S/w life cycle

Imp
Testing levels
- When software is developed as a product, testing of the product starts shortly after the start of its development.
- As development work progresses, the testing work also progresses and when development work is completed, the testing work is completed so on after.
- There are different levels during the process of testing
  i) Unit testing
  ii) Integration testing
  iii) System testing
  iv) User Acceptance Testing

Imp Importance of Level of Testing
i) To enhance the quality of software testing
ii) To produce more unified testing methodology applicable across several project.
iii) To introduce parallelism in the testing process as multiple test could be performed simultaneously.
iv) It is applicable across several software developments models.
v) The levels of testing have a hierarchical structure where higher levels assume successful and satisfactory completion of lower level tests.

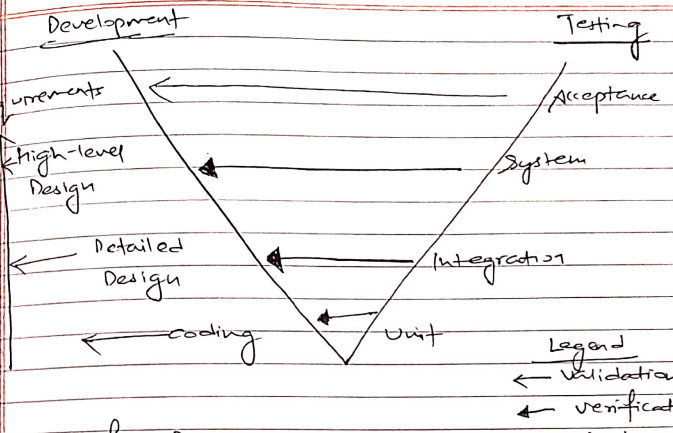


fig: Development and testing phases in the V-model

Chapter: 4

Testing throughout the Software Lifecycle:

4.1 Software Development Models:

There are six phases in SDLC model:

1. Requirement Gathering And Analysis
2. Design
3. Coding
4. Testing
5. Deployment
6. Maintenance.

There are various S/w development models or methodologies:

1. Waterfall Model.
2. V-model
3. Incremental model ~~Iterative model~~ / Iterative model
4. RAD model
5. Agile model
6. ~~Iterative model~~
7. Spiral model
8. Prototype model.

description at Lacie

4.2 Testing levels:

- When Software is developed as a product, testing of that product starts shortly after the start of the development.
- As development work progresses, the testing work also progresses and when development work is completed, the testing work is also completed soon after.
- There are different level during the process of testing

i) Unit testing
ii) Integration testing
iii) System testing
iv) User Acceptance testing.

(description of book)

Imp The importance of level of testing are:
- To identify missing areas and prevent overlap and repetition of testing between development lifecycle phases.
- To enhance the quality of software testing.
- To produce more unified testing methodologies applicable across several project.
- To introduce parallelism in the testing process as multiple test could be performed simultaneously.
- It is applicable across several software development models.
- The levels of testing have a hierarchical structure where high levels assume successful and satisfactory completion of lower level tests.

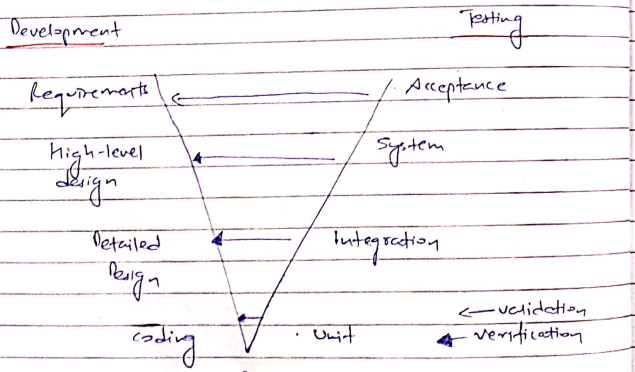


fig: Development & testing phase in the V-model
V: (verification & validation)

43) Test Types (functional testing
non-functional testing
Structural testing
Retesting
Regression testing)

Def

Functional Testing	Non-functional Testing
- It tells how well the system performs.	- It tells how well the system responds.
- It is based on client requirements.	- It is based on client expectations.
- It validates the behaviour of the application.	- It validates the performance of the application.
- Both are the part of system testing.	
- It includes: Unit Testing, integration testing, regression testing etc	- It includes: load/performance testing, stress/volume testing, security testing, installation testing etc.
- It shows how is our system doing.	- It shows how well is our system doing.
- It is given more priority	- It is given less priority.

- There is no use of non-functional testing if the product fails functional Testing.

Regression Testing	Retesting
- It is done to find out the issues which may get introduced because of any change or modification in the application.	- It is done to confirm whether the failed test cases in the final execution are working fine or not after the issues have been fixed.
- The purpose of ~ is that any new change in the application should NOT introduce any new bugs in existing functionality.	- The purpose of ~ is to ensure that the particular bug or issue is resolved and the functionality is working as expected.
- It has low priority.	- It has high priority.
- Test cases for ~ can be automated.	- Test cases for ~ cannot be automated.
- During ~ even the passed test cases are executed.	- During ~ only failed test cases are re-executed.
- Regression Testing is carried out to check for unexpected side effects.	- ~ is carried out to ensure that the original issue is working as expected.
- It consumes more time.	- It consumes less time.

Structural testing:

aka WBT -> explained before
& will be explained later too.

- testing of structure of the system or component.