Software

defination: Software engineering is a collection of techniques, methodologies and tools that helps with the production of

- a high quality sofware system
- with given budget
- before a given dealine

while change occurs.

'Bath-tub curve' :

Three phases of bathtub curve :

1) Early failure phase (Infant mortality) : This intial phase has a high failure rate. Caused by design flaws, Manufacturing defects, or errors in deployment process.

Relavence in S.E : This phase might be connnespond to the period after a software release where bugs are detected and fixed.

2) Normal Life Phase (useful Life) :

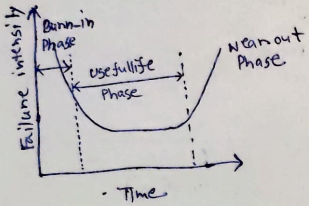
Defination Desenription : After the early failure are resolved, the Product enters a period of stability, where failure occure randomly and at a relatively low, constant rate. This Phase nepnasent the majority of the product lifecycle.

Relavence in S.E : During this Phase the software runs smoothly with occasional issues.

3) Wear-out Phase :

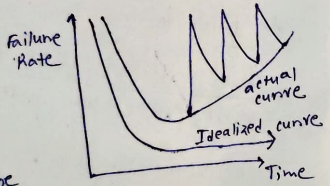
Description : Eventually, the failure rate increase again as the product age and components start to wear out.

Relevance : In software, this phase might occurs when the technology become outdates.



⇒ Wear vs Deterioration :

At its infant state, software has high failure rate as hardware. By time, After costomization & repairing the defects, it becomes idealize or gets into stedy state. Software defect may be happend by unfulfilled user demand, slow, bugs and many more. Here the idealized curve shows the idealized state and the actual curve shows the increased failure rate due to defect during software customization and modification. This causes Sofware ~~modification~~. So, software does not wear out, it just have deterioration.



deterioration

⇒ Characteristic of well-engineered software product :

1) Efficiency : The software should not make wasteful use of system resources.

2) Maintainability : It should be possible to evolve the software.

3) Dependability : It is the flexibility of the software that it did not cause any physical or economical injury within the system failure.

4) in-time : software should be developed well in time

5) within Budget : The software development cost should not overrun.

6) Functionality :

7) Adaptibility :

⇒ Software developer duties and responsibilities :

- Researching, designing, implementing software programs.
- Testing and evaluating new programs.
- Writing and implementing efficeient code.
- Developing quality assurance procedures.
- Maintaining and upgrading existing tools.
- Training users.

3 Q) Software engineering Phases:

Definition Phase

- Focuses on what (information engineering, software project planning)
- Focuses on how (Software design, Code generation)
- Focuses on change (connective maintainance, adaptive maintainance)

Q) Software development process model:

1) Big-Bang Model
2) code-and-fix "
3) waterfall model ①
4) V-model ② ~~prototyping model~~
5) Incremental model / ~~3~~ Iterative Enhancement model ③
6) RAD model ④
7) Agile model / ~~7~~ Evolutionary process model ⑤
8) Iterative model ~~8~~ Unified process model
9) Spinal model ⑥
10) Prototype model ⑦

Q) waterfall model: The following illustration is a representation of the different Phases of waterfall model.

Requirment analysis: All possible requirments of the system to be developed are captured in this phase and documented.

system design: The requirment specification from the first state phase are studied in this phase and system design is prepared.

Implementation: With the input from system design, the system is first developed in small programs called units. which are integrated in the next phase. Each unit is developed and tested for its functionality.

Integration and Testing: All the units developes in the ~~Integration~~ implemenation phase are integrated into a system after testing each unit.

Deployment of system: once the functional and non functional testing is done. the product is deployed in the costomer environment or release in the market.

Maintainance: There are some issue which come up in the client environment. To fix those issues Patches are released. Also to enhance the Product better version are released.

Application:

1) where the product defination is stable
2) where project is short
3) There are not ambigaous requirment
4) ~~Techo~~ where Technology is understood and not dynamic.

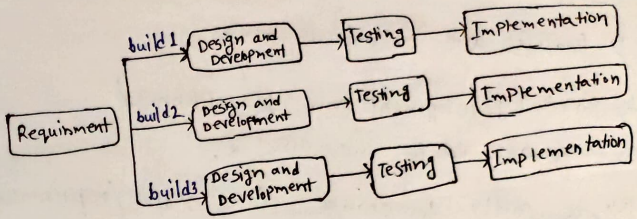
Advantage:

1) Simple and easy to understant ~~and~~ use.
2) Clearly define stages
3) Easy to arrange tasks
4) process and result are well documented.

Disadvantages:

1) No working software is produced until late during the life cycle.
2) High amount of risk and uncertainty
3) Not good model for complex and object-oriented projects.
4) Poor model for long on going ~~mode~~ projects.

Iterative model ~~and~~ or Incremental Model:



~~Requirment~~: In this incremental model the whole requirment is divided into ranies builds. During each Iteration the, the development module goes through Design and development, testing and Implementation phases. Each subsequent release of the module adds functions to the previous release.

~~As the~~

Application:

1) Requirment of the complete system are clearly defined and understood.
2) ~~A~~ A new technology is ~~beg~~ being used and being learnt by the development team while working on the project.
3) ~~The~~ There are some high-risks features and goals which may changes in future.

Advantages:

1) Results are obtained early
2) Parallel development can be planned
3) progress can be measured.
4) Less costly to change the scope/requirments.
5) Initial operation time is less.

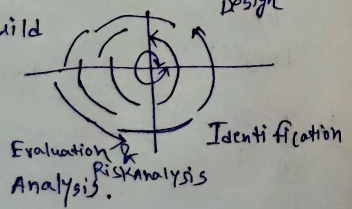
disadvantage:

i) More resources may be required.
ii) Not suitable for smaller projects.
iii) Management complexity is more.
iv) End of project may not be known which is a risk.

Spiral Model

The spiral model has four phases.

- Identification
- Design
- Construct or Build
- Evaluation and Risk Analysis.



i) Identification: The phase start with the gathering of system Requirments, subsystem Requirment. The phase also include understainding the system requirments by continious communication between the customer and the system analyst.

ii) Design: The design phase starts with the conceptual design in the baseline spiral and involves architectural design, Logical design of modules, Physical product design.

iii) Construct or Build: The construct phase refers to the production of actual software product. when a product is just thought of and the design is being developed, a POC (Proof of concept) is developed in this phase, to get customer feedback. Then in the subsequent spiral with higher clarity on requirments and design details a working model of software is produced.

iv) Evaluation and Risk analysis:- Risk analysis includes identifying, estimating, monitoring technical feasibility and management Risks.

Application

i) when budget constraint and risk evaluation is important
ii) For medium to high-risk project.
iii) Custome is not sure of their requirments.
iv) significant changes are expected in the product during the development cycle.

Advantage:

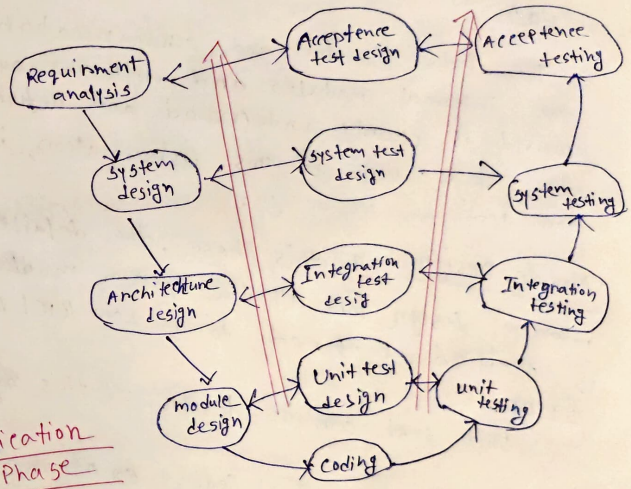
i) User see the system early
ii) Changing requirment can be possible.
iii) Better Risk management.
iv) Allows extensive use of prototype.

disadvantage:

i) End of the project may not be known early.
ii) Management is more complex
iii) Not suitable for small and low risky project.
iv) process is complex.

V-Model: Varification and validation model.

The v-model is an extension of water fall model and based on the association of testing phase for each corrosponding development phase.



i) Requirment ~~anal~~ analysis: This is the first phase of development cycle where the customers requirments are understood. The acceptence test design planning is done at this stage.

ii) System Design: Once you have the clear and detailed Product Requirment, it is the time for design the complete system.
The system test plan is developed based on the system design.

Architectural Design: Architectural specifications are understood and Design in this Phase. This is also referred as high level Design (HLD).

The Data Transfer and communication of the internal modules and with the outside world is cleanly understood and define in this stage. With this information, integration test can be design.

Module Design: In this Phase, the detailed internal design for all the system module is specified, referred to as Low level Design. (LLD).

Unit test can be design at this stage.

Coding Phase: The actual coding of the system module design is taken up in the coding Phase. The best suitable programming language is decided.

The coding is performed based on the coding guidline and standards.

Validation Phase

Unit testing: is designed in the module design phase. And executed on validation Phase. Unit testing is the tesing at code level and helps eleiminate bugs at an early stage.

Integration testing: is associated with the architectural design phase. Integration tests are performed to test the communication of internal modules within the system.

System testing:

System testing is directly associated with the system design phase.

Acceptance testing: Acceptance testing is associated with the business requirement analysis phase.

Application

Same as waterfall model.

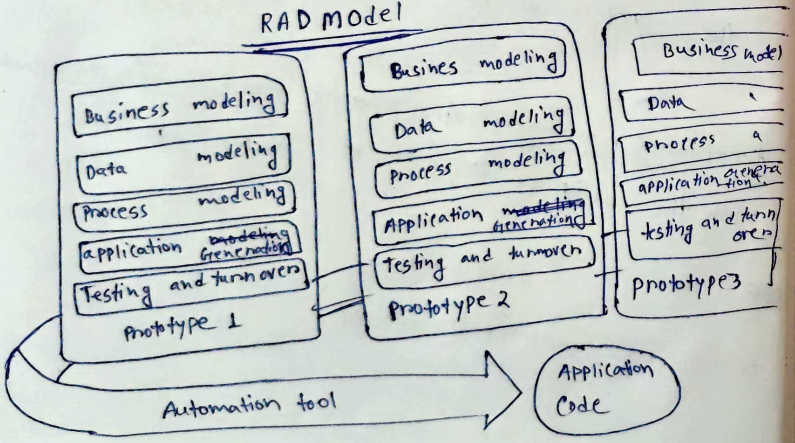
∴ following are the some suitable scenarios:-

i) Requirments are well defined, cleanly documented.
ii) product defination is stable.
iii) Technology is not dynamic.
iv) The project is short.

DisAdvantages:

i) High Risk and uncertainty.
ii) Not good for complex and object oriented projects.
iii) Poor model for long on-going projects.
iv) No working software is produce until late during the life cycle.

RAD Model



Business Modeling: A complete business analysis is performed to find the vital information about business, how and when the is the information processed.

Data Modeling: The information that gathered in Business modeling phase is review and analyzed to form set of object vital for the business.

Process Modeling: process description for adding, deleting, retrieving and modifying data object are given.

Application Generation: The actual system is built and coding is done by using automation tools to convert process and data model into actual prototype.

Testing and Turnover: The overall testing time is reduced as the prototypes are independently tested during every iteration.

RAD VS SDLC

SDLC

i) The traditional SDLC follows a rigid process
ii) High Requirment and gathering before the coding stants.
iii) The requirments are needed before the project stant from customer.
iv) changes of requirment is rigid and it may not be feasible for project.

RAD

i) The RAD Model focuses on iterative and incremental delivary
ii) This results in rapid delivary to customers

iii) Changes of requirment is easier than SDLC
iv) Reducing the risk of non-conformence.

Application

i) It should be used if there is high availability of designers.
ii) It should be used if the budget permits the use of automation tools.
iii) Should be used when there is chance of requirment changing.
iv) Should be used where working prototypes are to be presented in short period of time.

Advantage

i) Changing requirment is possiable.
ii) Reduce development time
iii) progress can be measured.
iv) increase reusability of components.
v) suitable for shorten development time requiring project.

Disadvantage

i) Requires highly skilled developers/designers
ii) High dependency on modeling skills.
iii) Inapplicable to cheapen projects.
iv) Shuitable for project requiring shorten development time.
v) Poor model for long ongoing project.

Software Prototyping

The software prototyping refers to software application prototype which display functionality of the product under development.

i) Basic Requirment Identification: Understanding the basic Product requirment in terms of User Interface.
ii) Developing initial Prototype: The initial prototype is developed in this stage. And user interfaces are provide.
iii) Review of prototype: The prototype presented to the customer. And geting feedback from customer.
iv) Revice and enhence the prototype: Feedback and review comment are discussed during this stage.

Types

Throw away / Rapid prototyping: Once the actual Requirments are understood, the prototype is discarded and the actual system is developed with a much clearn understanding of user requirments.

Evolutionary Prototyping: where the prototype is continiously refined based on user feedback until it evolves into the final Product.

Incremental prototyping: refers to building multiple prototypes of various subsystem and then integrating all the available prototype to form a complete system

Extreme prototyping: used in web development domain.

Application

i) It is useful in developing system where high level of user interaction.

ii) System which need users to fill out forms or go through various screens before the data is processed.

iii) Software that involves too much data processing.

Advantages

i) Reduce time and cost and time.

ii) defect can be detected much easier.

iii) Quicker user feedback is available leading to better solution.

iv) The user get better understanding of the system.

disadvantages

i) User may confused in in the prototype and actual system.

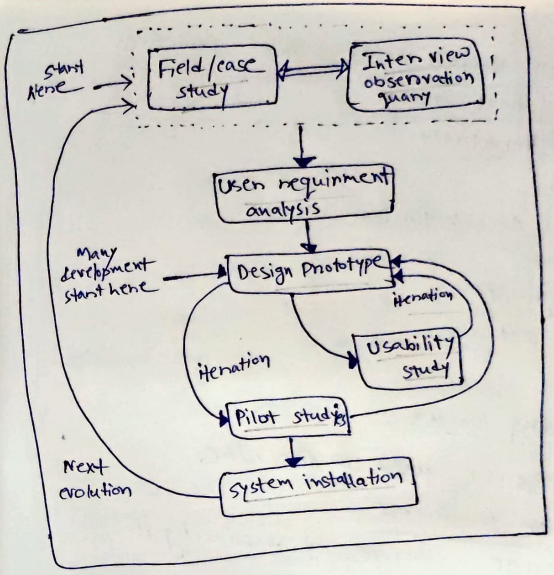
ii) Increase the complexity of the system.

iii) Insufficient Requirment analysis

iv) Too much dependency on the prototype.

Evolutionary process model

* Its a combination of Iterative and Incremental model users are able to get access to the product at the end of each cycle.



Evolutionary Process Model

Application :

i) It is used in Large project.

ii) used when customer wants to use core features

iii) It also used in object oriented software development.

Advantage :

i) user can experience Partially developed system

ii) It reduces the errors.

iii) Customer satisfaction is high.

disadvantage

i) some time it is hard to decide the problem into several version

ii)

Rational Unified process Best practices

6 fundamental Best Practic

i) Develope software Iteratively
ii) Manage Requirments
iii) Use Component-Based Architectures
iv) visually model software
v) verify software quality
vi) control changes to software

The Building Blocks

The Building Blocks are used to describe

i) What should be produced
ii) Who is incharge Change of producing it.
iii) how production will take place.
iv) When production is complete.

Workflows (when) further deride into 6 core engineering workflows :

i) Business modeling workflow
ii) Requirment workflow
iii) Analysis and design workflow
iv) Implementation workflow
v) Test workflow
vi) Deployment workflow.

Chapter = 3

1) Requirment of certification [it means why we don't need certification]

i) How often will developer need certification to keep Pace with new technology.

2) How will any certification ~~not~~ address the issue Like, Analytical and Logical reasoning, programming aptitude, Positive attitude.

3) Certification can not guarantee high quality product.

4) ~~Whether~~ Whether we go for certified developer or certified processes.

5) Types of certification :

People
↳ Industry specific

process
↳ Industry specific

Product
↳ for the customer directly and helps to select a particular product.

Certification of Persons

- Some of the most respected certification SQA (software quality assurance) professionals ~~are~~ include:
  - i) certified software quality Analyst (CSQA)
  - ii) certified software tester (CSTE)
  - iii) certified u project Manager (CSPM)
- Some company specific certification are also popular like microsoft office specialist (MOS) in word, Excel, and powerpoint

Certification of process:

- The most popular process certification approaches are
  - i) ISO 9000
  - ii) SEI-CMM (Software Engineering Institute_capability Maturity model)
- One should always be suspecious about the quality of end product. however, the certification reduces the possibility of • Poor quality product.
- Any types of process certification helps to produce good quality of software product.

increase possibility of good quality of software product.