Ian Somerville.
8th edition.

15 march 21.

UNIT - 1

Software Engineering - software engineering is concerned with developing software products which can be of 2 types 1. Bespoken product 2. Generic product

Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stage of system specifications and maintaining the system after it has gone into use.

Engineering discipline applies to theories, methods, tools where these are appropriate to discover the solution.
All aspects → involving project management of with development tools, methods and theories to support software production for producing high quality software.

Computer Science is concerned with the theory & methods which underly computer and software system whereas software engineering is concerned with practical aspect of problems of producing software.

Software Engineering is concerned with all aspects of development and evolution of a complex system where, software plays a major role. System engineering is concerned with hardware, policy, processes, design and system development as well as software engineering.

Project management

Req. Design Coding Testing Release Maintenance

Process

23 March 21

Software Process - A software process is a set of activities & associated results that produce a software product.

There are 4 fundamental activities

1. Software specifications - This activity defines the software functionality and constraints on its operation.

2. Software development - Involves production of software meeting the specifications.

3. Software validation - This activity involves validating the software to ensure it does what the customer wants.

4. Software evolution - This involves evolution of software to meet changing customer needs.

Each process organizes the tasks and activities in different ways with different levels of details. Each process produces a different work products. If process is inappropriate it will reduce the quality or usefulness of the resulting work product & final product.

Software Process Model. - Is a simplified description of a software process activity or activities presented from a particular perspective. Process models may include part of the software process, eg. software product & roles of people involved in software engineering.

Ex - 1. Work flow model
2. Data flow model / Activity model
3. Role / Action model.

1. Workflow model - describes the sequence of activities in the process providing information about inputs, outputs, dependencies and human actions.

2. Dataflow model - Represents the process as a set of activities each of which carries out some data transformation. This shows how I/p are transformed into O/p through activities, people or computer.

3. Role / action model - This model describes roles of people involved in the software process & activities they are responsible for.

different products

Repeatable form ⇒ Assurance

General software Development Models or Paradigm.

1. Waterfall model
2. Iterative model / Evolutionary model
3. Component based / Reuse

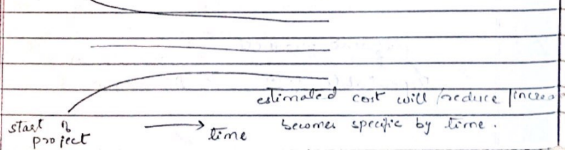
1. Waterfall model represents software development in following phases.
   1. Requirement
   2. development
   3. Validation.

The product is refined with customer inputs to produce a final system satisfying customer needs.

2. Iterative model - This approach interleaves the activities of specification, development & validation. An initial system is rapidly developed from abstract situation and further refined with customer inputs to produce a product that satisfies the customer.

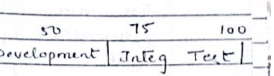
3. Component Based - Technique assumes that a part of a system already exists. System development focuses on integrating the inputs rather than developing them.

25 March 21

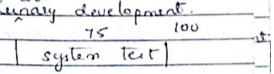


prototype development → Exploratory Development.
→ Throwaway prototype

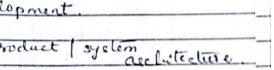
Water fall model.



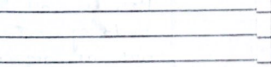
Iterative development / Evolutionary development.



component base s/w Development.



Dev & evolution



Software Engineering cost.

There is no simple answer to define software cost in a software project. This is since the process is dependent on the software development model and the type of software being used and characteristics of such software, ...

S/w is invisible & clarity of requirement estimate & final product characteristics is visible towards the later part of development process.

General rule for customised software requirement provided by the customer → 40-20-40
Req. code T&t
Design

off the shelf product

For mass customized software (COTS) the specification effort is shorter & system testing effort is longer. to make the product useable over large range of customer.

Software Engeneering Methods

Software Engeneering Method is a structured approach to software development which aims to facilitate the production of high quality software in a cost effective manner.

Provide all the technical "how to's" for building software and encompassing models such as requirements, design, program construction, Testing and maintenance. These methods are based on a set of basic principles that govern each area of the Technology and to identify the basic functional components.

Pg 8312 evolution of s/w eng methods.

Dennis
function oriented
Rambaugh
object oriented
Booch

Software development

No software engg practices → Results in undependable & unpredictable software product

Software engg practices → Good software product

Product characteristics / behaviour.

- Maintainable
- Dependable
- efficiency
- Usability

Attributes of good software - software have set of associated attributes that reflect the quality of that software. The attributes are not directly concerned with the software does but they reflect its behaviour why it executing & the structure of the program.

Ex - Response time,

Specific set of attributes of a good software are. 1. Maintainable
2. Dependable
3. Efficiency
4. Usability

Key challenges to software engineering.

1. Heterogeneous challenges.
2. Delivery challenge.
3. Trust challenges.

1. Increasingly systems are required to operate as distributed systems across networks that include different types of computers & with different kinds of support system. Also it may be necessary to integrate s/w with legacy systems. The heterogeneity challenge is the challenge of developing techniques for building dependable s/w that is flexible enough with heterogeneity.

2. It a challenge of shortening the delivery time for large & complex systems without compromising system quality.
Techniques

3. It to develop trust that demonstrates s/w can be trusted by user.

Professional & Ethical Responsibility.

s/w engg is carried out within a legal & social framework that limit the freedom of engg. s/w engg job involves application of skill & wider responsibility i.e. involving application of technical skills. They must also behave in an ethical & morally responsible way if they have to be respected as professionals. some of the professional responsibilities are.

1. confidentiality
2. competence
3. intellectual property rights
4. computer misuse
5. privacy

page 21

Process Iteration [Iterative model
spiral model. (Risk based)

Software models

1. Waterfall model consists of the following stages
   1. Requirement definition
   2. System & software design
   3. Implementation & unit testing.
   4. Integration & system testing
   5. operation & maintainence.

This is also called as classical lifecycle model of software development
model is based on conventional engineering technique where each phase doesn't start until the previous phase is completed

(1) Requirement def - consists of system engineering & analysis where requirements are gathered at system level

(2) Software Requirement analysis
This focuses on gathering software requirements & understand the nature of software prg to be build. Implementing the software requirements required functions

behaviour, performance and interface.

(2) System & software design - involves identifying & describing the fundamental software system & abstraction & their relationship. This phase focuses on 4 distinct attributes of a program
1. software architecture
2. Data structure
3. Interface implementation
4. Procedural details (algorithm)

Design phase translates software requirements into representation of software that can be assessed for quality.

(3) Implementation & unit testing - During this phase software design is realised as a set of using a software language. set of program units.

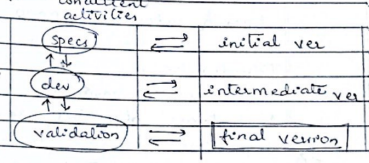
unit testing verifies that each unit meets its specifications.

(4) Integrating & system testing - Individual prog units are integrated and tested as a complete system to ensure software requirements have been met.

(5) operation & maintainance - During this phase the system is installed to put into practical use. Maintainance phase involves activities that uncovered correct errors that were not discovered in early life cycle, to improve & enhance the software services.

2. Evolutionary Development

1. Exploratory development
2. Throwaway prototype



Evolutionary development is refered to as prototyping.
This model develops.
software product based on defined specifiers defined by customers & exposing this to user & refining the versions until an adequate s/w is developed.

Exploratory development.
Developer works with the customer to explore requirements & deliver the final system.
The system is enhanced by adding new features & final system is evolved.

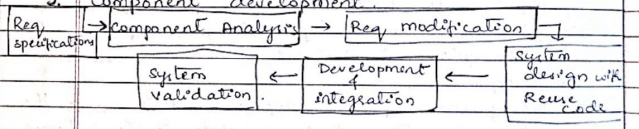
Throwaway
The purpose of this approach is to evolve the system definition that is close to the customers requirements.
The model focuses on experimenting with customer requirements that are poorly understood

Advantages
1. more effective than waterfall model

2. Specifications can be developed incrementally.
3. User if the developer develop better understood of requirements.
4. Suitable for small to medium sized (100000 lines of codes to 500000.

Disadvantages
1. Process is invisible
2. Customer objectives are used to develop requirements.
   a. specification, development & validation happens concurrently.
   b. Deliverables cannot be identified and measurement of program is difficult
   c. less cost effective.
3. Poor system structure.
4. special tools & techniques may be incompatible

3. Component development.



1. Requirement specification
2. Component Analysis
3. Requirement modification
4. System ~~validation~~ design with reuse
5. Development of integration
6. System validation

4. The generic process model for reuse oriented development consists of following stages. (1-6) in ← page.

1. Requirement specification - same as waterfall

2. Component Analysis - Based on the requirement spec, a search is made for components to implement specifications. Generally there is no exact match with ~~expert~~ reuse to functionality.

3. Req modification - Analyses the requirements to suit the requirements of functionality. In situation where modification is not possible, a search is made to find more suitable component.

4. Sys design & reuse - Framework of system design is developed or an existing design is reused. If reusable components are not available, new design will be evolved

5. Development & Integration - Available reusable s/w is used with appropriate changes. In situations where software cannot be reused, new software is developed & integrated to system.

6. System validation - system is validated to ensure that it performs as per customer wants

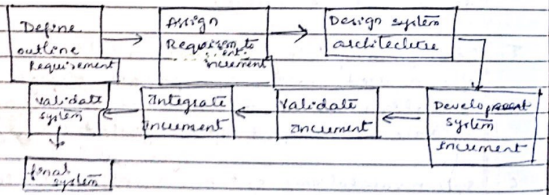
Advantages
1. Reduces cost
2. Reduces risk analysis
3. Faster delivery.

Disadvantages
1. some control over the s/w structure is lost as the new version may not be under configuration

1 April 21

Process Iteration (Incremental development) - Is an iterative development model which combines the approaches of waterfall model & evolutionary model. Customer identifies the system services and prioritize the services. Series of increments are identified with delivery functionality. The delivery is based on the priority assigned to the services. During the next step the requirement for the services to be delivered during the first increment.

The increment is developed during the defined process. The customer reviews the version & clarifies the requirements for the next increment.



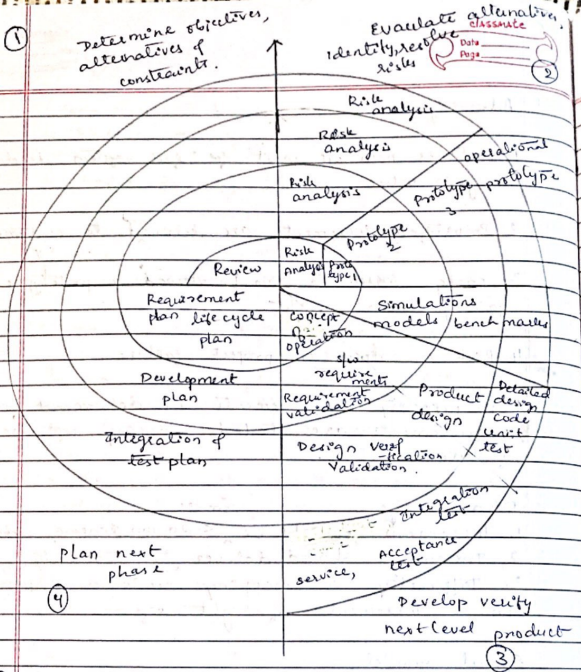
New increments developed are integrated into the existing increment to improve the system functionality with each new increment.

Advantages
1. Customer need not wait for full system development to gain value
2. Prioritized increments are delivered to customer at
3. early increments can be viewed as prototypes for gaining better insight
4. Risk is down on project failure.

High priority requirements are delivered first & requirements are tested reducing software failure.

Disadvantages
1. Increment should be small < 20,000 lines of code
2. Increment should deliver functionality
3. Difficulty in mapping customer requirements on to increments of right size.

Spiral model
Spiral model was proposed by Boehm in 1988. The model represents the sequence of software process activities as a spiral. Each loop in a spiral represents a phase of the software process. Each loop in a spiral is divided into 4 sectors
1. objective setting
2. Risk assessment & reduction
3. Development & validation
4. Planning.



(1) Objective setting - During this phase, project specific objectives are specified
- Process & product constraints are identified
- Detailed management plant is drawn
- Project risks are identified
- Alternative strategies depending on risks are planned.

2. Risk assessment & reduction - for each project risk is identified & a detailed analysis is carried out. steps are taken to reduce risk

3. Development & validation
A suitable development model for the system is selected based on specific criteria like interface risks relating to
(1) Interface risk
(2) safety risk
(3) subsystem integration

(4) Planning - The review of the project status is performed & decision is made to continue on further loop of the spiral.

Risks are potentionally adverse circumstances that may impact development process and quality of the product.

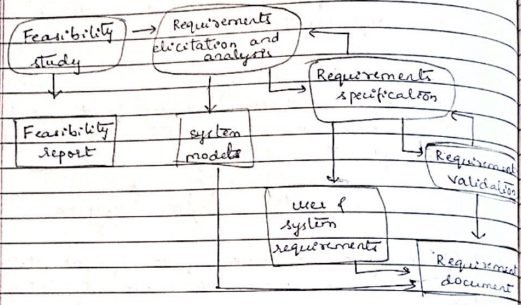
Process Activities (meta level process)

1. understandable
2. simple
3. unambigious
4. complete

There 4 fundamental process activities
1. Specification
2. Development
3. validation
4. Evolution.

Software engineering / Requirement process / specification

Software specifications -



1. Feasibility study
2. Requirements elicitation & analysis.
3. Requirement specification
4. Requirement validation
5.

1. Feasibility study - This phase estimates wheather identified user need may be satisfied using current software & hardware. Decision is made wheather to go ahead with existing details or need to extract more details from customer

2. Req elicitation & Analysis - Elicitation is the process of extracting or deriving the system requirements through observation of existing system or through discussion with potentional users.
Analysis is the process of discovery, refinement of models

3. Req specification - activity of translating the info gathered during the analysis activity into a document that defines the set of user requirements / system requirements

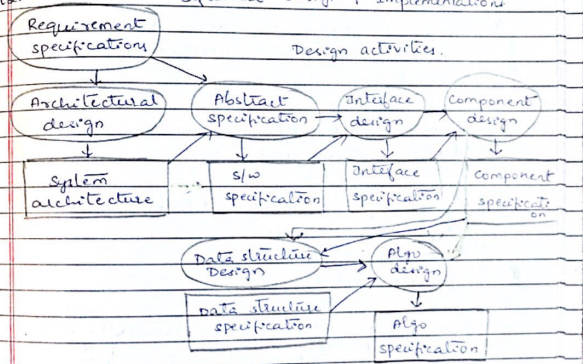
4. Req validation - Checks the requirements for realism, consistency, completeness of errors are discovered

5. Req document - Document is prepared & signed off with customers

Req specification

6 April 21

Software design & implementation



S/w design describes the s/w structure to be implemented representating

1. s/w architecture
2. Abstract specifications
3. Interface design
4. Component design
5. Data structure design
6. Algo design.

1. Architectural design - It is the sub system making up the system & their relationship is identified along with the information flow.

2. Abstract specification - Provides information on the abstract specification subsystem services & constraints under which it must operate.

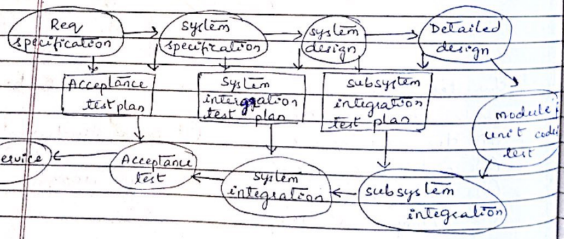
3. Interface (specification) design - This provides information on each subsystem interface with other subsystem

4. Component design - services are allocated to different components & interface of these components are design.

5. Data structure design - Provides information on the data structures used.

6. Algo design - Provides details of the algo used to provide the services

III Software Validation



validation involves both verification & validation.

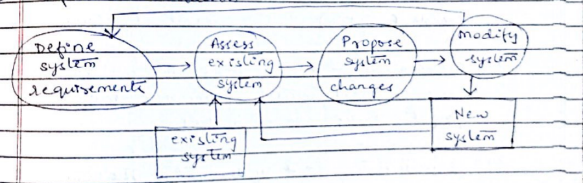
The stages in the testing process are

1. Component (or unit) testing : Individual components are tested to ensure that they operate correctly.

2. System testing : components are integrated to make up the system. This process is concerned with finding errors that results from unanticipated interaction b/w components & component interface problems.

3. Acceptance testing - This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system customers rather than with simulated test data.

IV. Software evolution



Rational unified process (RUP)

→ Phase iteration

Inception	Elaboration	construction	Transition

RUP recognise the conventional process models present a single view of the process. In contrast, the RUP describes 3 perspectives

1. Dynamic perspective that shows the phases of the model over time.

2. A static perspective that shows the process activities that are enacted.

3. A practice perspective that suggests good practices to be used during the process.

1. Inception – * It establishes business case
   * Identifies all external entities (people & system) that interact & define the system

2. Elaboration – * understands the problem domain & defines architectural framework project plan & project risks are established.

3. construction – * System design, programming, Testing working, sln system & documentation are performed

4. Transition – * The concern is to move the system from the development community to the user community along with documentation &

operational system.

Computer aided software engineering (CASE) technology provides sln process support by automating some process activities & providing information about the sln been developed

Activities that can be automated are

1. Development of graphical system model in requirements specifications & design.

2. Data dictionaries holding information about entity & relationship in the design.

3. Generation of user interface from the graphical interface description that is created interactively.

4. Program debugging through the provision of information about an executing program.

Fig 4.16

Pg 87