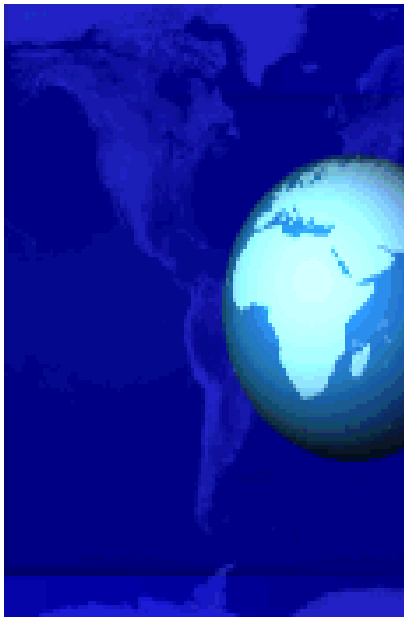


Software Engineering

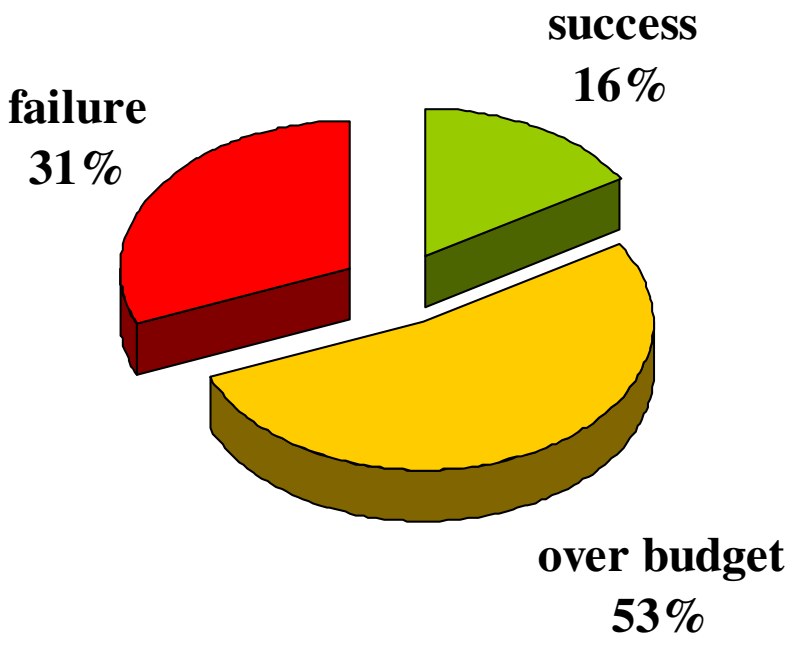
Why Software Engineering ?

- Change in nature & complexity of software
- Concept of one "guru" is over
- We all want improvement

Ready for change

The Evolving Role of Softwar

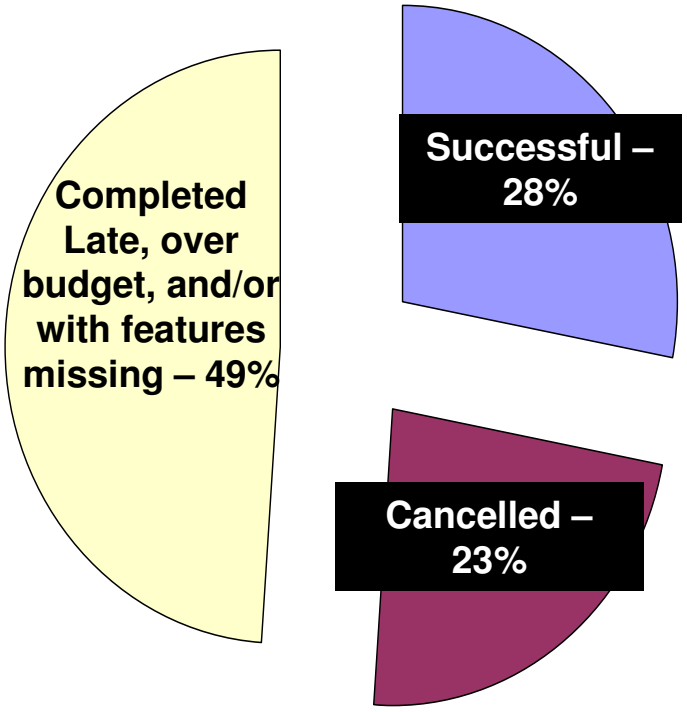
- Software industry is in Crisis!



Source: The Standish Group International, Inc. (CH

The Evolving Role of Software

This is the SORRY state of Software Engineering Today!

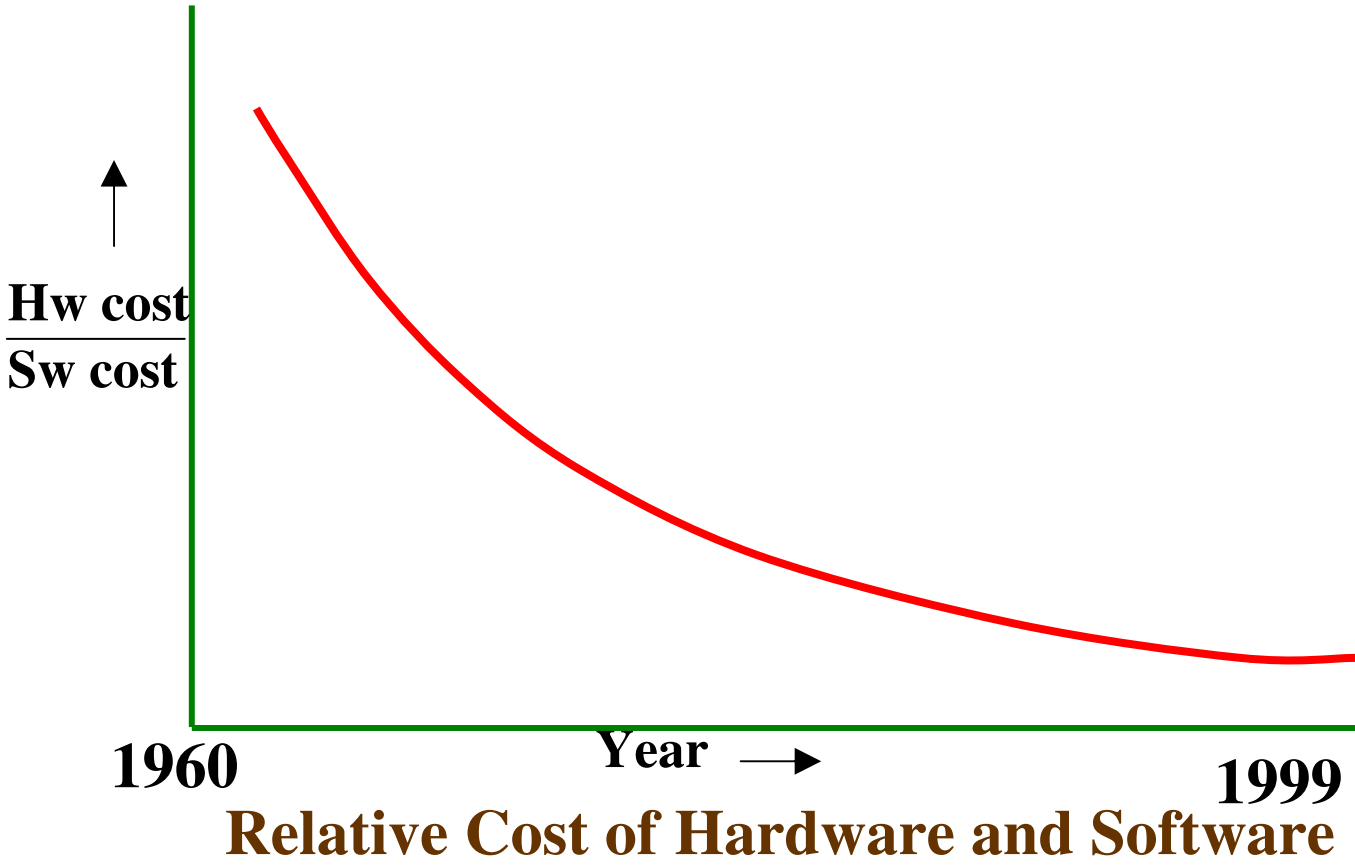


- **Data on 28,000 projects completed in 2000**

The Evolving Role of Software

As per the IBM report, "31%of the pro
cancelled before they are completed, 53
run their cost estimates by an average o
and for every 100 projects, there are 94 re

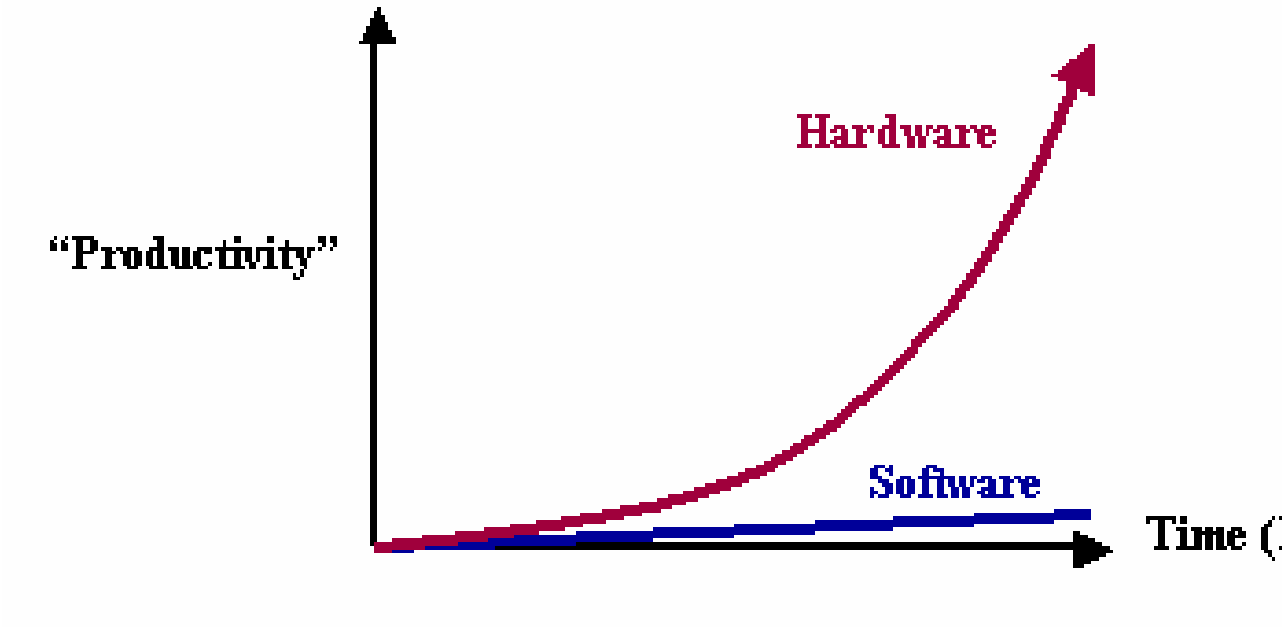
The Evolving Role of Software



Relative Cost of Hardware and Software

The Evolving Role of Software

- Unlike Hardware
  - Moore's law: processor speed/memory capac every two years



The Evolving Role of Software

***Managers* and *Technical Persons* are asked:**

- ✓ Why does it take so long to get the program finishe
- ✓ Why are costs so high?
- ✓ Why can not we find all errors before release?
- ✓ Why do we have difficulty in measuring progress d development?

Factors Contributing to the Software

- Larger problems,
- Lack of adequate training in software eng
- Increasing skill shortage,
- Low productivity improvements.

Some Software failures

Ariane 5

It took the European Space Agency **10 years and $7 billion** to produce Ariane 5, a giant rocket capable of hurling a pair of three-ton satellites into orbit with each launch and intended to give Europe overwhelming supremacy in the commercial space business.

The rocket was destroyed after 39 seconds of its launch, at an altitude of two and a half miles along with its payload of four expensive and uninsured scientific satellites.

Some Software failures

When the guidance system's own computer tried to convert one piece of data the sideways velocity of the rocket from a 64 bit format to a 16 bit format; the number was too big, and an overflow error resulted after 36.7 seconds. When the guidance system shutdown, it passed control to an identical, redundant unit, which was there to provide backup in case of just such a failure. Unfortunately, the second unit, which had failed in the identical manner a few milliseconds before.

Some Software failures

Y2K problem:

It was simply the ignorance about the adequacy or otherwise of using only last two digits of the year.

The 4-digit date format, like 1964, was shortened to 2-digit format, like 64.

Some Software failures

The Patriot Missile

- First time used in Gulf war
- Used as a defense from Iraqi Scud missiles
- Failed several times including one that killed 28 US soldiers in Dhahran, Saudi Arabia

Reasons:

A small timing error in the system's clock accumulated to the point that after 14 hours, the tracking system was no longer accurate. In the Dhahran attack, the system had been operating for more than 100 hours.

Some Software failures

The Space Shuttle

Part of an abort scenario for the Shuttle requires fuel dumps to lighten the spacecraft. It was during the second of these dumps that a (software) crash occurred.

...the fuel management module, which had performed one dump and successfully exited, restarted when recalled for the second fuel dump...

Some Software failures

A simple fix took care of the problem.. programmers decided to see if they could come systematic way to eliminate these generic sorts c the future. A random group of programmers ap system to the fuel dump module and other modul

Seventeen additional, previously unknown surfaced!

Some Software failures

Financial Software

Many companies have experienced failures accounting system due to faults in the software failures range from producing the wrong infor the whole system crashing.

Some Software failures

Windows XP

- Microsoft released Windows XP on October 2
- On the same day company posted 18 compatibility patches on the website for b compatibility updates, and enhancements.
- Two patches fixed important security holes.

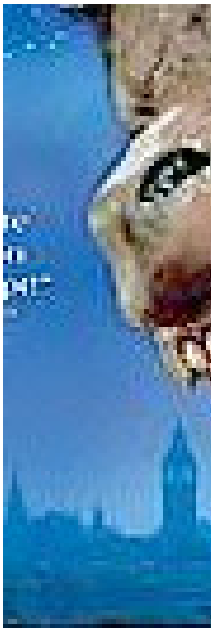
This is **Software Engineering.**

"No Silver Bullet"

The hardware cost continues to decline drastically.

However, there are desperate cries for a silver bullet something to make software costs drop as rapidly as computer hardware costs do.

But as we look to the horizon of a decade, we see no silver bullet. There is no single development, either in technology or in management technique, that by itself promises even one order of magnitude improvement in productivity, in reliability and in simplicity.



"No Silver Bullet"

The hard part of building software is the specification, testing of this conceptual construct, not the labour of rep and testing the correctness of representation.

We still make syntax errors, to be sure, but they a compared to the conceptual errors (logic errors) in m That is why, building software is always hard and there no silver bullet.

While there is no royal road, there is a path forward.

Is reusability (and open source) the new silver bullet?

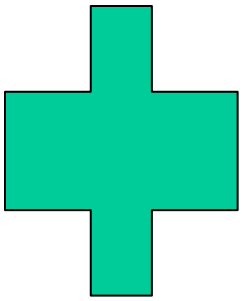
"No Silver Bullet"

The blame for software bugs belongs to:

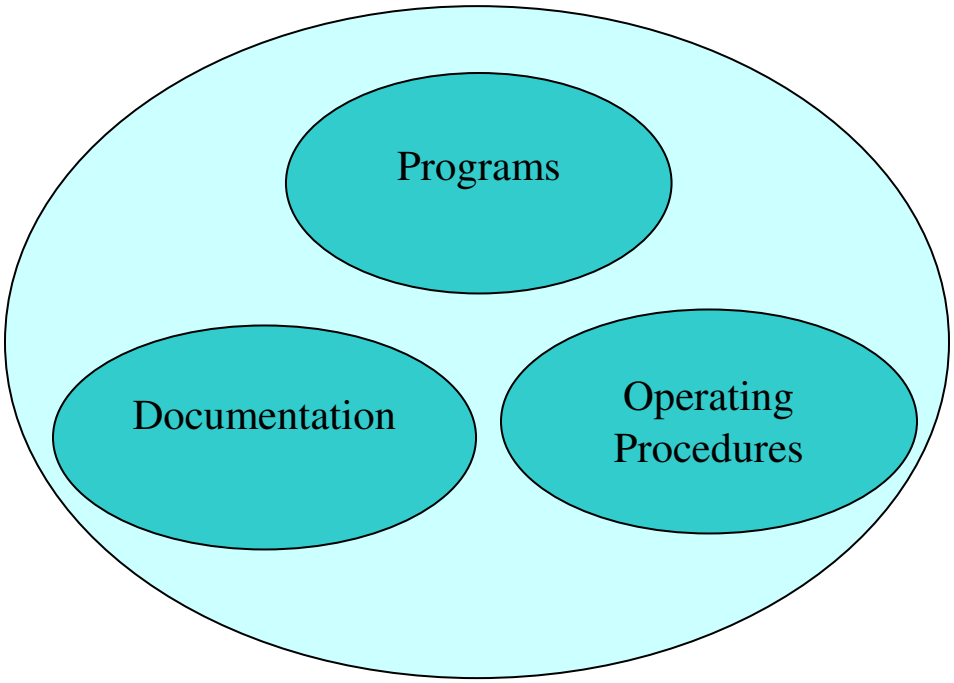
- Software companies
- Software developers
- Legal system
- Universities

What is software?

- **Computer programs** and **asso**
  documentation



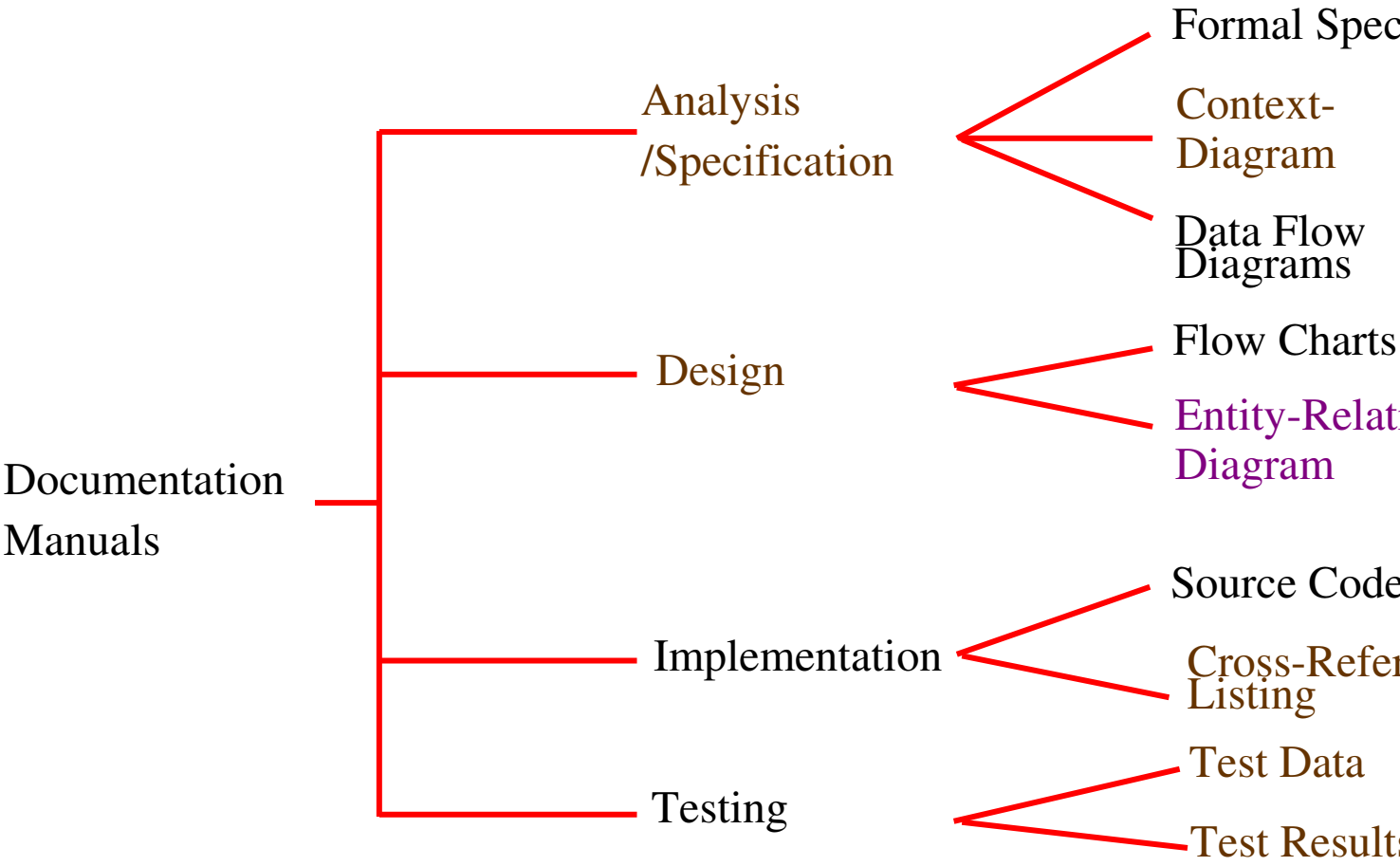
What is software?



Software=Program+Documentation+Operating Proc

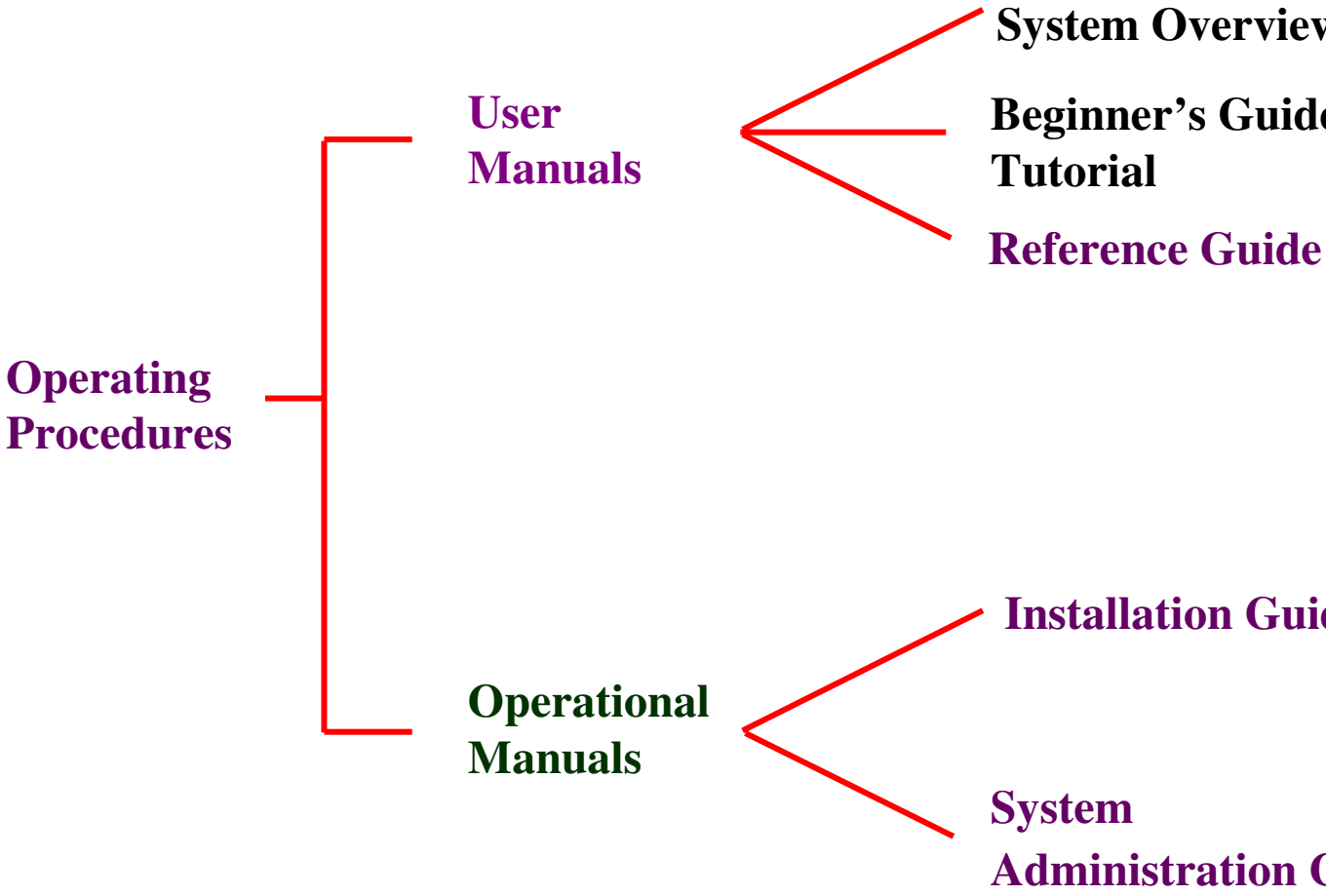
Components of software

Documentation consists of different types of ma



List of documentation manuals

Documentation consists of different types of ma



List of operating procedure manuals.

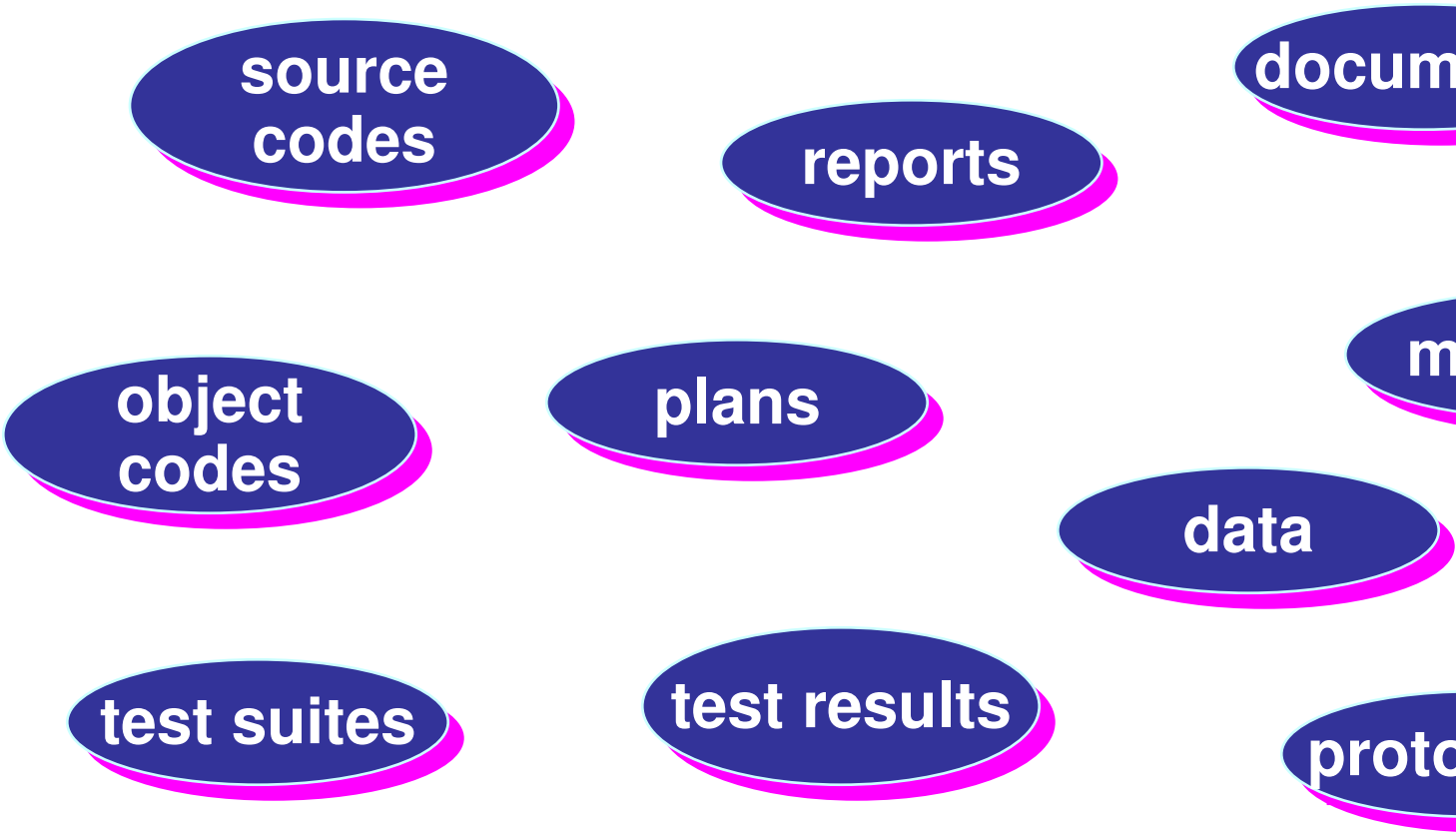
Software Product

- **Software products** may be developed for a customer or may be developed for a general ma

- **Software products** may be
  - **Generic** - developed to be sold to a range o customers
  - **Bespoke** (custom) - developed for a single customer to their specification

Software Product

Software product is a product designated for delivery to the user



What is software engineering?

Software engineering is an engineering discipl
is concerned with all aspects of software product

Software engineers should

- adopt a systematic and organised approach
  work
- use appropriate tools and techniques depen
  - the problem to be solved,
  - the development constraints and
- use the resources available

What is software engineering?

At the first conference on software engineering in 1968, defined software engineering as "The establishment sound engineering principles in order to obtain ec developed software that is reliable and works efficien machines".

Stephen Schach defined the same as "A discipline whose production of quality software, software that is delivere within budget, and that satisfies its requirements".

Both the definitions are popular and acceptable t However, due to increase in cost of maintaining softwar is now shifting to produce quality software that is m delivered on time, within budget, and also satisfies its re

Software Process

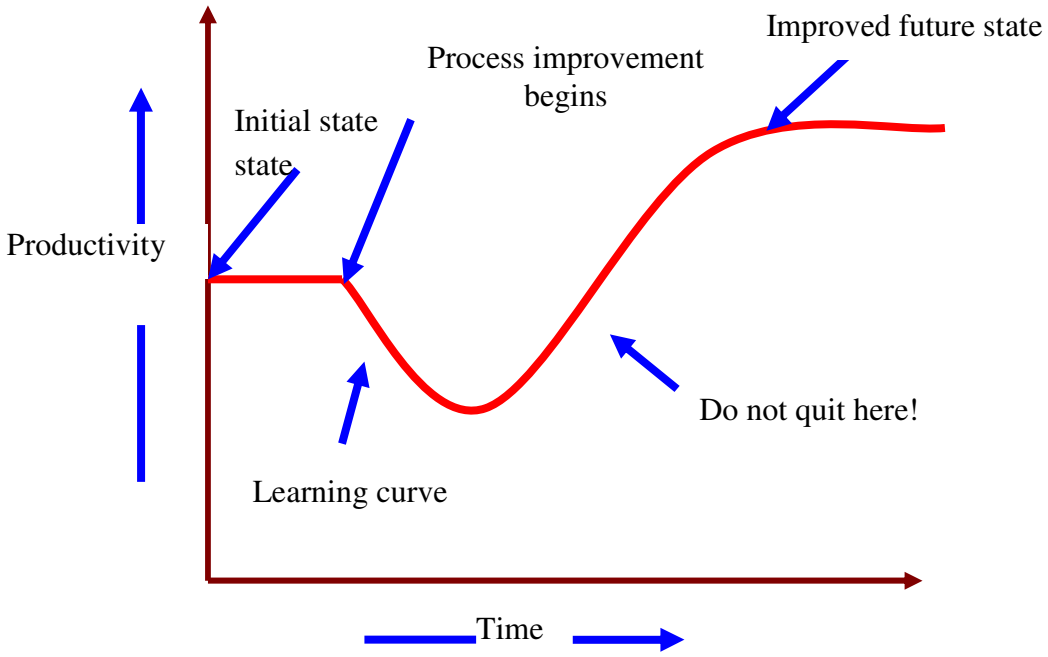
The software process is the way in which we
software.

Why is it difficult to improve software process ?

- Not enough time
- Lack of knowledge

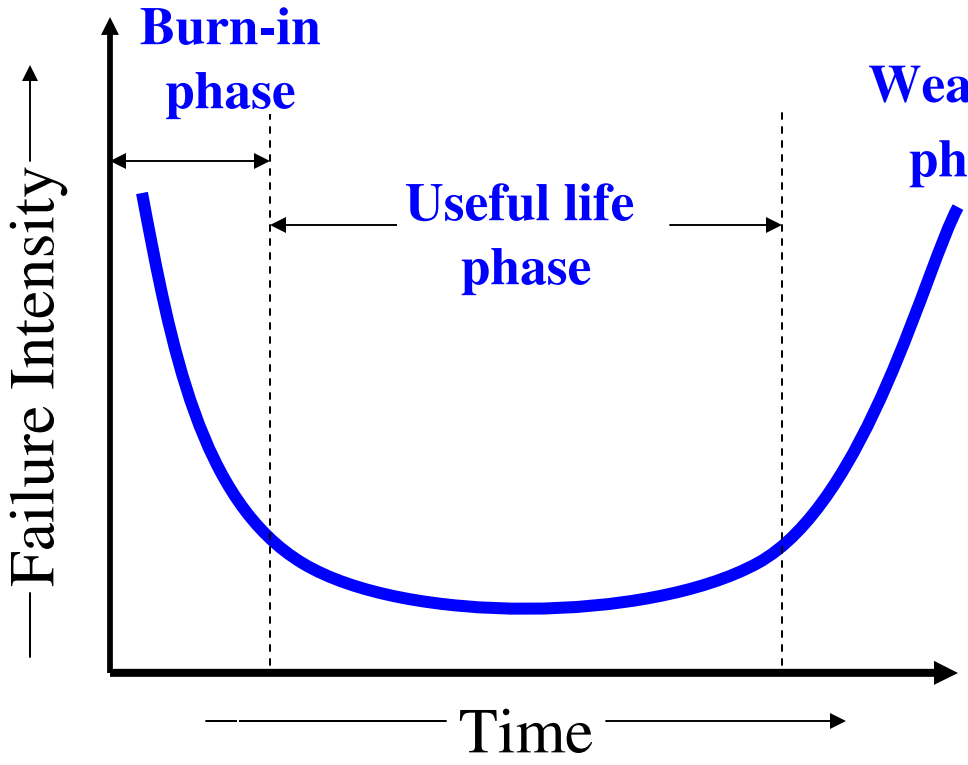
Software Process

- Wrong motivations
- Insufficient commitment



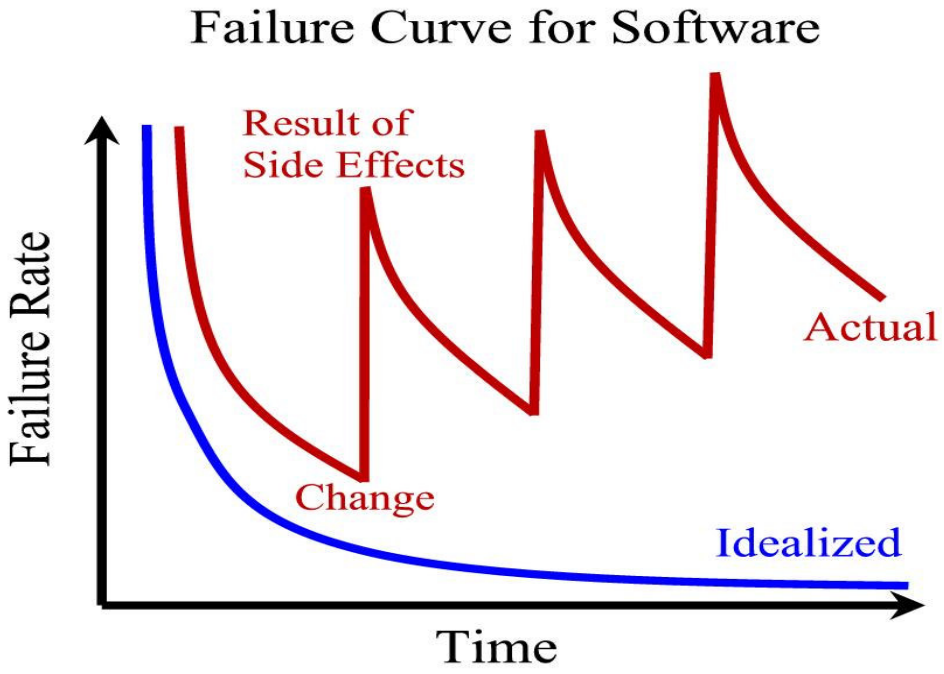
Software Characteristics:

✓Software does not wear out.



Software Characteristics:

- ✓ Software is not manufactured
- ✓ Reusability of components
- ✓ Software is flexible

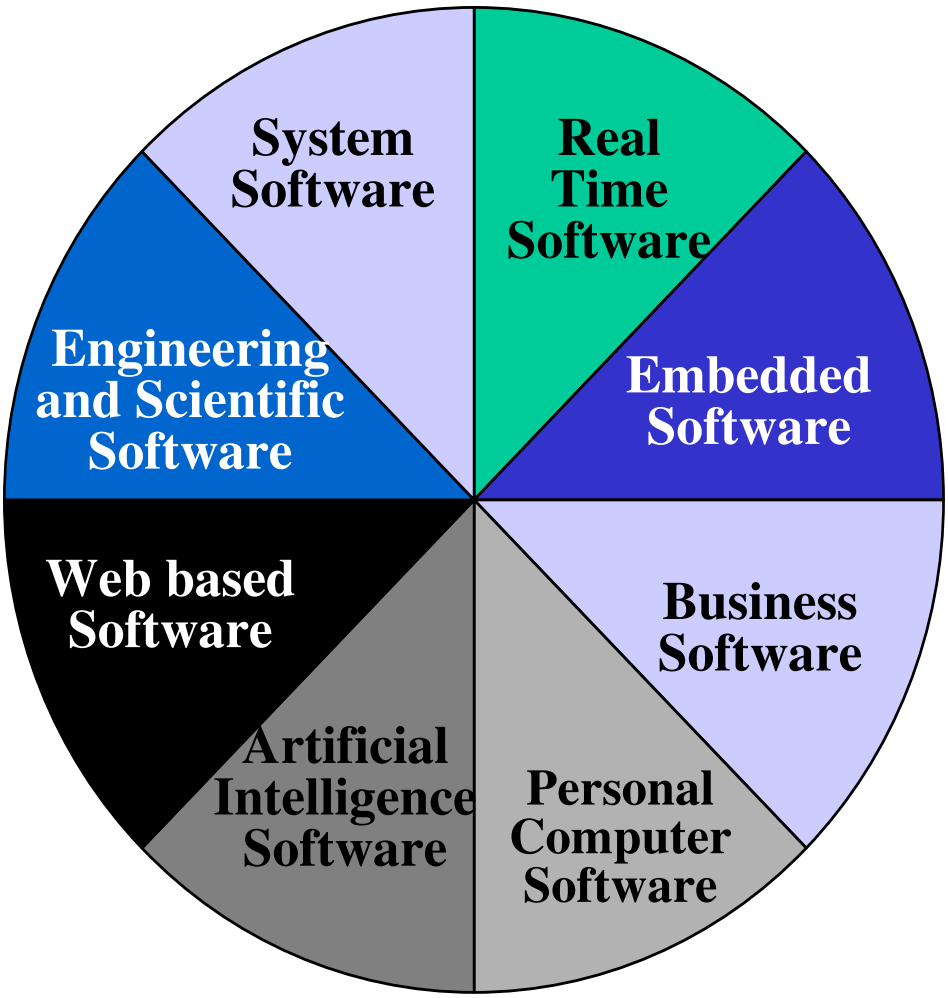


Software Characteristics:

Comparison of constructing a bridge vis-à-vis writing a

Sr. No	Constructing a bridge	Writing a prog
1.	The problem is well understood	Only some parts of th understood, others are not
2.	There are many existing bridges	Every program is different a special applications.
3.	The requirement for a bridge typically do not change much during construction	Requirements typically ch phases of development.
4.	The strength and stability of a bridge can be calculated with reasonable precision	Not possible to calculate program with existing metho
5.	When a bridge collapses, there is a detailed investigation and report	When a program fails, the re unavailable or even deliberat
6.	Engineers have been constructing bridges for thousands of years	Developers have been writi for 50 years or so.
7.	Materials (wood, stone,iron, steel) and techniques (making joints in wood, carving stone, casting iron) change slowly.	Hardware and software chan

The Changing Nature of Softwa



The Changing Nature of Softwa

Trend has emerged to provide source code to the customer and organizations.

Software where source codes are available are k as open source software.

Examples

Open source software: LINUX, MySQL, PHP, Open of Apache webserver etc.

Software Myths (Management Perspe

Management may be confident about goo
standards and clear procedures of the com

But the taste of any food item is in the eating; not in the Recipe !

Software Myths (Management Perspe

Company has latest computers and st
the-art software tools, so we shouldn't
about the quality of the product.

The infrastructure is only one of the several factors that determine the quality of the product!

Software Myths (Management Perspe

Addition of more software specialists with higher skills and longer experienc bring the schedule back on the track!

Unfortunately, that may further delay the schedule!

Software Myths (Management Perspe

Software is easy to change

The reality is totally different.

Software Myths (Management Perspe

Computers provide greater reliability than the devices they replace

This is not always true.

Software Myths (Customer Perspect

A general statement of objectives is sufficient to get st
the development of software. Missing/vague requirer
easily be incorporated/detailed out as they get concreti

If we do so, we are heading towards a disaster.

Software Myths (Customer Perspect

Software with more features is better software

Software can work right the first time

Both are only myths!

Software Myths (Developer Perspec

Once the software is demonstrated, the job is d

Usually, the problems just begin!

Software Myths (Developer Perspect

Software quality can not be assessed testing.

However, quality assessment techniques should be used through out the software development life cycle.

Software Myths (Developer Perspect

The only deliverable for a so
development project is the tested code.

Tested code is only one of the deliverable!

Software Myths (Developer Perspect

Aim is to develop working programs

Those days are over. Now objective is to develop good quality maintainable programs!

Some Terminologies

- Deliverables and Milestones

Different deliverables are generated during software d
The examples are source code, user manuals, operatin
manuals etc.

The milestones are the events that are used to ascertain
the project. Finalization of specification is a milestone. C
design documentation is another milestone. The mil
essential for project planning and management.

Some Terminologies

- Product and Process

Product: What is delivered to the customer, is called a
may include source code, specification document
documentation etc. Basically, it is nothing but a set of
only.

Process: Process is the way in which we produce softwa
collection of activities that leads to (a part of) a product. A
process is required to produce good quality products.

If the process is weak, the end product will undoubtedly
an obsessive over reliance on process is also dangerous.

Some Terminologies

- Measures, Metrics and Measurement

A measure provides a quantitative indication of dimension, size, capacity, efficiency, productivity or r some attributes of a product or process.

Measurement is the act of evaluating a measure.

A metric is a quantitative measure of the degree to whic component or process possesses a given attribute.

Some Terminologies

- Software Process and Product Metrics

Process metrics quantify the attributes of software process and environment;

whereas product metrics are measures for the software pr

Examples

Process metrics: Productivity, Quality, Efficiency etc.

Product metrics: Size, Reliability, Complexity etc.

Some Terminologies

- Productivity and Effort

Productivity is defined as the rate of output, or productio
effort, i.e. the output achieved with regard to the tim
irrespective of the cost incurred.

Hence most appropriate unit of effort is Person Mo
meaning thereby number of persons involved for speci
So, productivity may be measured as LOC/PM (lin
produced/person month)

Some Terminologies

- Module and Software Components

There are many definitions of the term module. They ra
module is a FORTRAN subroutine" to "a module
Package", to "Procedures and functions of PASCAL
"C++ Java classes" to "Java packages" to "a module
assignment for an individual developer". All these de
correct. The term subprogram is also used sometimes
module.

Some Terminologies

"An independently deliverable piece of functionality access to its services through interfaces".

"A component represents a modular, deployable, and part of a system that encapsulates implementation and ex of interfaces".

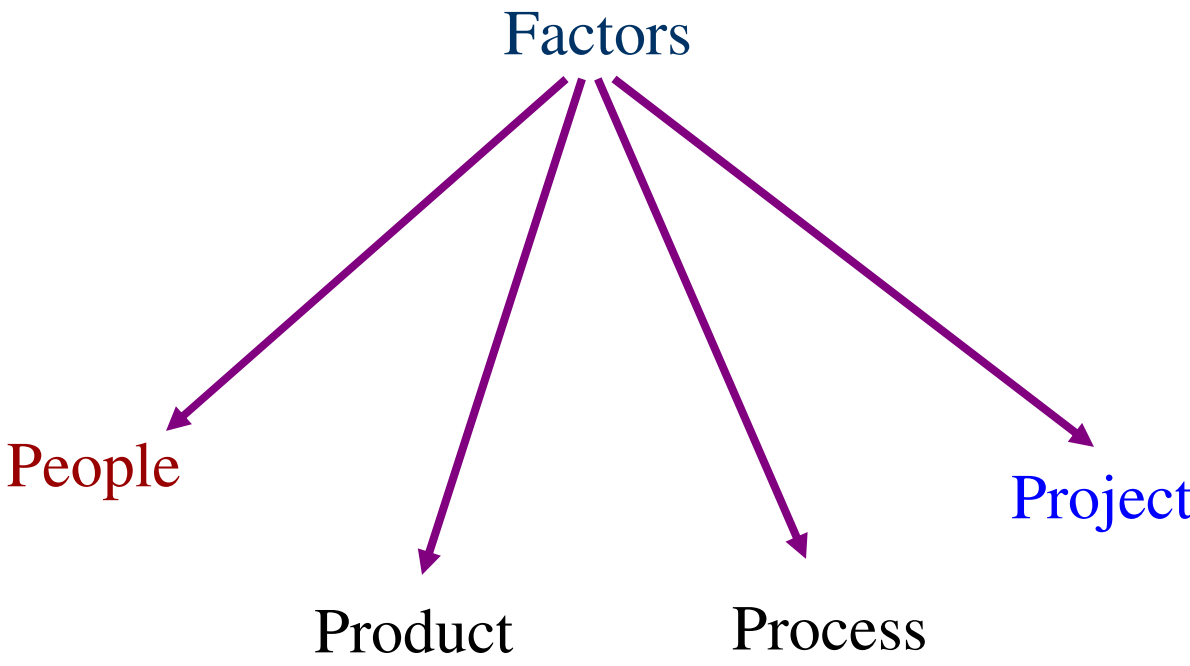
Some Terminologies

- Generic and Customized Software Products

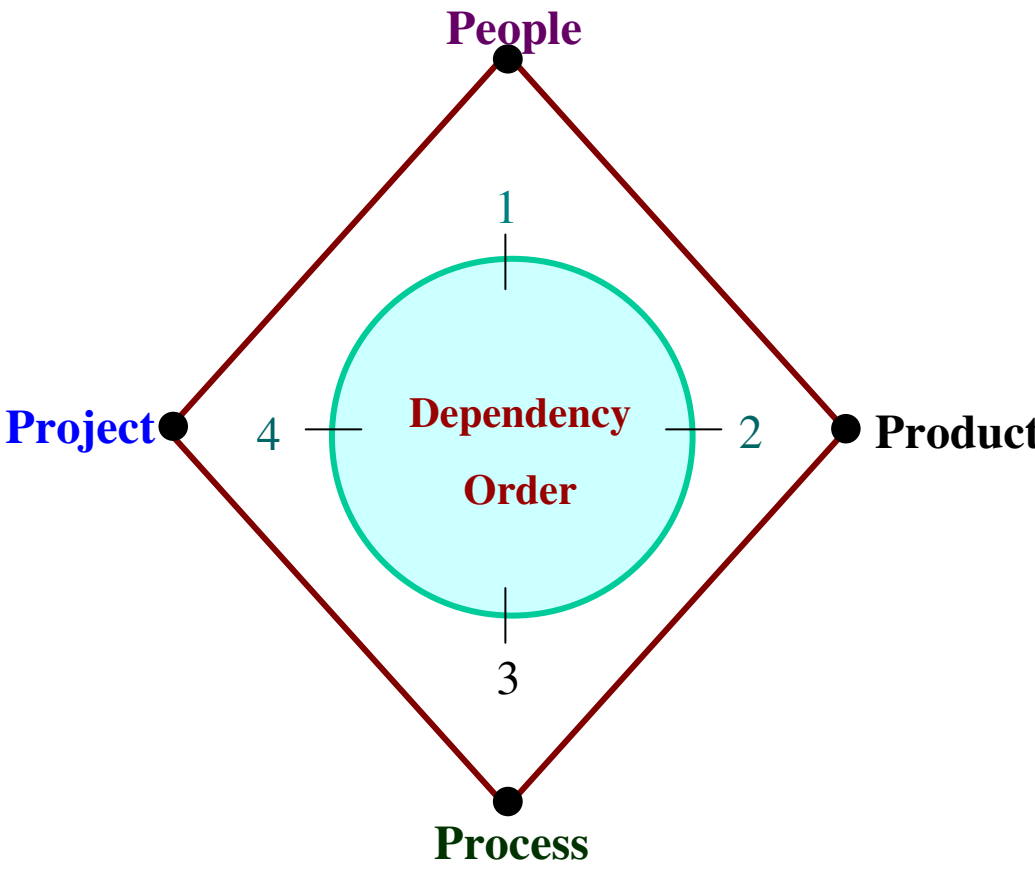
Generic products are developed for anonymous customers is generally the entire world and many copies are expected Infrastructure software like operating system, compilers word processors, CASE tools etc. are covered in this categ

The customized products are developed for particular The specific product is designed and developed as pe requirements. Most of the development projects (80%)come under this category.

Role of Management in Software Deve



Role of Management in Software Deve



Multiple Choice Questions

Note: Select most appropriate answer of the following qu

1.1 Software is

(a) Superset of programs (b) subset of programs
(c) Set of programs (d) none of the above

1.2 Which is NOT the part of operating procedure manuals?

(a) User manuals (b) Operational manual
(c) Documentation manuals (d) Installation manuals

1.3 Which is NOT a software characteristic?

(a) Software does not wear out (b) Software is flexible
(c) Software is not manufactured (d) Software is always

1.4 Product is

(a) Deliverables (b) User expectations
(c) Organization's effort in development (d) none of the above

1.5 To produce a good quality product, process should be

(a) Complex (b) Efficient
(c) Rigorous (d) none of the above

Multiple Choice Questions

Note: Select most appropriate answer of the following q

1.6 Which is not a product metric?

(a) Size (b) Reliability

(c) Productivity (d) Functionality

1.7 Which is NOT a process metric?

(a) Productivity (b) Functionality

(c) Quality (d) Efficiency

1.8 Effort is measured in terms of:

(a) Person-months (b) Rupees

(c) Persons (d) Months

1.9 UML stands for

(a) Uniform modeling language (b) Unified modeling

(c) Unit modeling language (d) Universal modelin

1.1 An independently deliverable piece of functionality providing ac its services through interface is called

(a) Software measurement (b) Software composit

(c) Software measure (d) Software compone

Multiple Choice Questions

Note: Select most appropriate answer of the following qu

1.11 Infrastructure software are covered under

(a) Generic products
(b) Customized product
(c) Generic and Customized products
(d) none of the above

1.12 Management of software development is dependent on

(a) people
(b) product
(c) process
(d) all of the above

1.13 During software development, which factor is most crucial?

(a) People
(b) Product
(c) Process
(d) Project

1.14 Program is

(a) subset of software
(b) super set of softwar
(c) software
(d) none of the above

1.15 Milestones are used to

(a) know the cost of the project
(b) know the status of t
(c) know user expectations
(d) none of the above

Multiple Choice Questions

Note: Select most appropriate answer of the following question

1.16 The term module used during design phase refers to

(a) Function (b) Procedure

(c) Sub program (d) All of the above

1.17 Software consists of

(a) Set of instructions + operating system

(b) Programs + documentation + operating procedures

(c) Programs + hardware manuals (d) Set of programs

1.18 Software engineering approach is used to achieve:

(a) Better performance of hardware (b) Error free software

(c) Reusable software (d) Quality software proc

1.19 Concept of software engineering are applicable to

(a) Fortran language only (b) Pascal language only

(c) 'C' language only (d) All of the above

1.20 CASE Tool is

(a) Computer Aided Software Engineering (b) Component Aided Softwar

(c) Constructive Aided Software Engineering (d)Computer Analysis Softv

Exercises

1.1 Why is primary goal of software development now producing good quality software to good quality maintainable s

1.2 List the reasons for the "software crisis"?Why are CA normally able to control it?

1.3 "The software crisis is aggravated by the progress technology?" Explain with examples.

1.4 What is software crisis? Was Y2K a software crisis?

1.5 What is the significance of software crisis in referenc engineering discipline.

1.6 How are software myths affecting software process? Exp help of examples.

1.7 State the difference between program and software. Why ha and documentation become very important.

1.8 What is software engineering? Is it an art, craft or a science

Exercises

1.9 What is aim of software engineering? What does the software engineering discuss?

1.10 Define the term "Software engineering". Explain the maj between software engineering and other traditional engineering

1.11 What is software process? Why is it difficult to improve it'

1.12 Describe the characteristics of software contrasting characteristics of hardware.

1.13 Write down the major characteristics of a software. Ill diagram that the software does not wear out.

1.14 What are the components of a software? Discuss how a so from a program.

1.15 Discuss major areas of the applications of the software.

1.16 Is software a product or process? Justify your answer with

Exercises

1.17 Differentiate between the following

(i) Deliverables and milestones (ii) Product and proces

(iii) Measures, metrics and measurement

1.18 What is software metric? How is it different fr measurement

1.19 Discuss software process and product metrics with the help

1.20 What is productivity? How is it related to effort. What effort.

1.21 Differentiate between module and software component.

1.22 Distinguish between generic and customized software pro one has larger share of market and why?

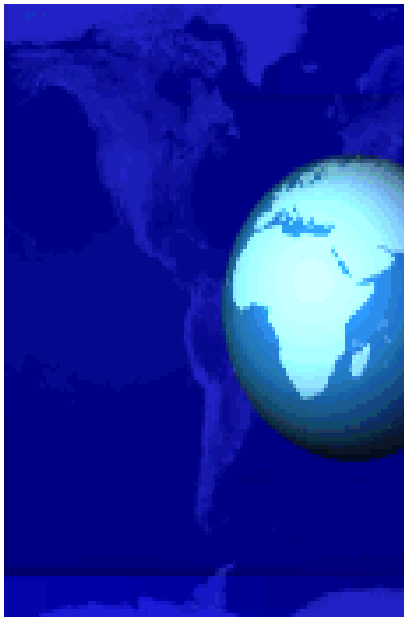
1.23 Is software a product or process? Justify your answer with

Exercises

1.23 Describe the role of management in software development of examples.

1.24 What are various factors of management dependency development. Discuss each factor in detail.

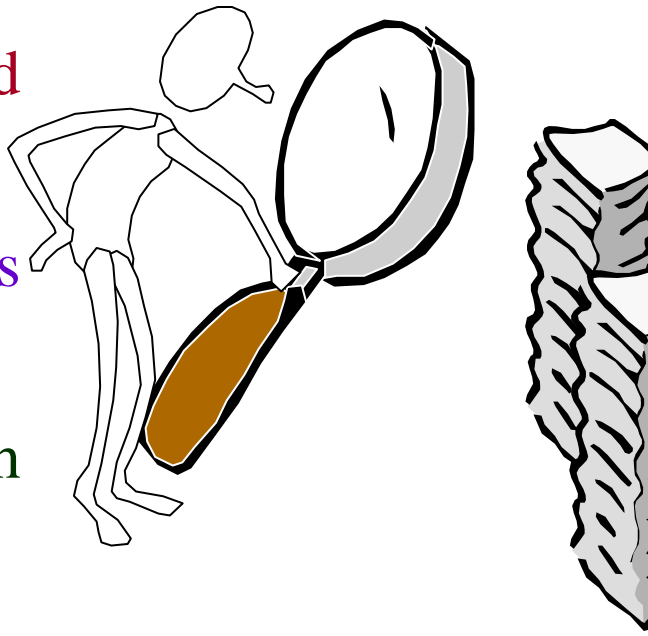
1.25 What is more important: Product or process? Justify your



Software Certification

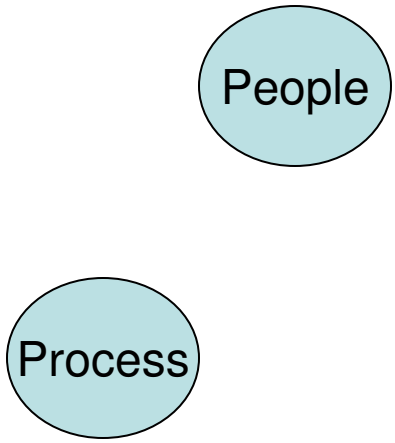
Software Certification

- What is certification?
- Why should we really need it?
- Who should carry out this activity?
- Where should we do such type of certification?



Software Certification

To whom should we target



- People
- Process
- Product

We have seen many certified developers (Microsoft c
Cisco certified, JAVA certified), certified processes (lik
CMM) and certified products.

There is no clarity about the procedure of software certif

Requirement of Certification

Adam Kalawa of Parasoft has given his views on certific

"I strongly oppose certification of software developers. will bring more harm than good to the software indu further hurt software quality by shifting the blar software. The campaign for certification assumes that developers cause software problem and that we c software quality by ensuring that all developers have stamp of approval. However, improving quali improving the production process and integrating in to that reduce the opportunity for introducing defec product"

Requirement of Certification

- How often will developers require certification to kee new technologies?
- How will any certification address the issues like fu of computer science, analytical & logical programming aptitude & positive attitude?
- Process certification alone cannot guarantee h product.
- Whether we go for certified developers or certified pr

Can independent certification agency provide a f field for each software industry??

Types of Certification

- People
  - Industry specific
- Process
  - Industry specific
- Product
  - For the customer directly and helps to select product

Certification of Persons

The individual obtaining certification receives the follov

- Recognition by peers
- Increased confidence in personal capabilities
- Recognition by software industry for professional ac
- Improvement in processes
- Competences maintained through recertification

Certification is employees initiated improvement proc
improves competence in quality assurances methods & te

Certification of Persons

Professional level of competence in the principles & p
software quality assurance in the software industr
achieved by acquiring the designation of:

- Certified Software Quality Analyst (CSQA)
- Certified Software Tester (CSTE)
- Certified Software Project Manager (CSPM)

Some company specific certifications are also very po
Microsoft Office Specialist (MOS) certifications in W
and PowerPoint.

MOS is far best known computer skills certifi
administrator.

Certification of Processes

The most popular process certification approaches are:

- ISO 9000
- SEI-CMM

One should always be suspicious about the qualit product, however, certification reduces the possibilit quality products.

Any type of process certification helps to produce go and stable software product.

Certification of Products

- This is what is required for the customer.
- There is no universally accepted product ce scheme.
- Aviation industry has a popular certification "R 178B".
- The targeted certification level is either A, B, C, D,
- These levels describe the consequences of a poten of the software : catastrophic, hazardous seve minor or no effect.

Certification of Products

DO-178B Records

Software Development Plan

Software Verification Plan

Software Configuration Management Pla

Software Quality Assurance Plan

Software Requirements Standards

Software Design Document

Software Verification Test Cases & Produ

Certification of Products

DO-178B Documents

- Software Verification Results
- Problem Report
- Software Configuration Management Recor
- Software Quality Assurance Records

DO-178B certification process is most demanding at hi

Certification of Products

DO-178B level A will:

1. Have largest potential market
2. Require thorough labour intensive preparation
   the items on the DO-178B support list.

DO-178B Level E would:

1. Require fewer support item and
2. Less taxing on company resources.

Certification of Products

We don’t have product certification in most of the area
(real time operating system) is the real-time operating s
certification & marked as “LinuxOS-178”.

The establishment of independent agencies is a viable

Third Party Certification for Comp base Software Engineering

Weyukar has rightly said "For Component based Development (CBO) to revolutionalize software de developers must be able to produce software significan and faster than they otherwise could, even as the resulti meets the same sort of high reliability standards while be maintain".

Bill council has also given his views as "Currently, the evidences that component based software engineering revolutionizing software development, and lots of reason otherwise. I believe the primary reason is that the comm showing how to develop trusted components".

Third Party Certification for Comp base Software Engineering

Contractor:

- Gives the standard
- Directs any variations in specification
- Define patterns
- Allowable tolerances
- Fix the date of delivery

Third party certification is a method to ensure software c conform to well defined standards, based on this ce trusted assemblies of components can be constructed

Third party certification is based on UL 1998, 2nd ed., U for safety for software in programmable component.

Exercises

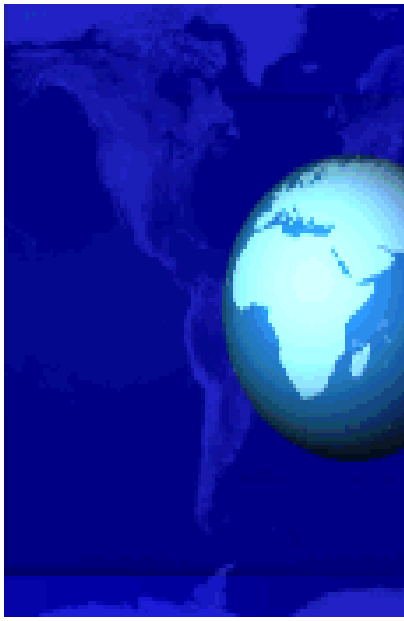
10.1 What is software certification? Discuss its importance in scenario of software industry.

10.2 What are different types of certifications? Explain the s each type & which one is most important for the end user.

10.3 What is the role of third party certification in component b engineering? Why are we not able to stabilize the component b engineering practices.

10.4 Name few person specific certification schemes. Which popular & why?

10.5 Why customer is only interested in product certification product certification techniques with their generic applicability.



Software Life Cycle Models

Software Life Cycle Models

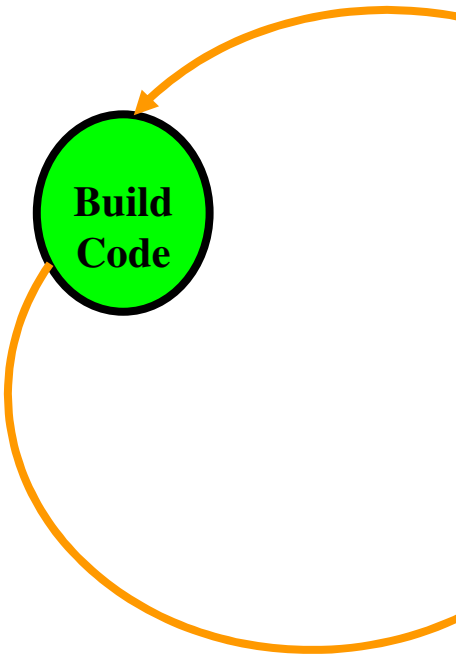
The goal of Software Engineering is to
models and processes that lead
production of well-documented main
software in a manner that is predictable.

Software Life Cycle Models

"The period of time that starts when a software product
and ends when the product is no longer available f
software life cycle typically includes a requirement pl
phase, implementation phase, test phase, installation an
phase, operation and maintenance phase, and sometime
phase".

Build & Fix Model

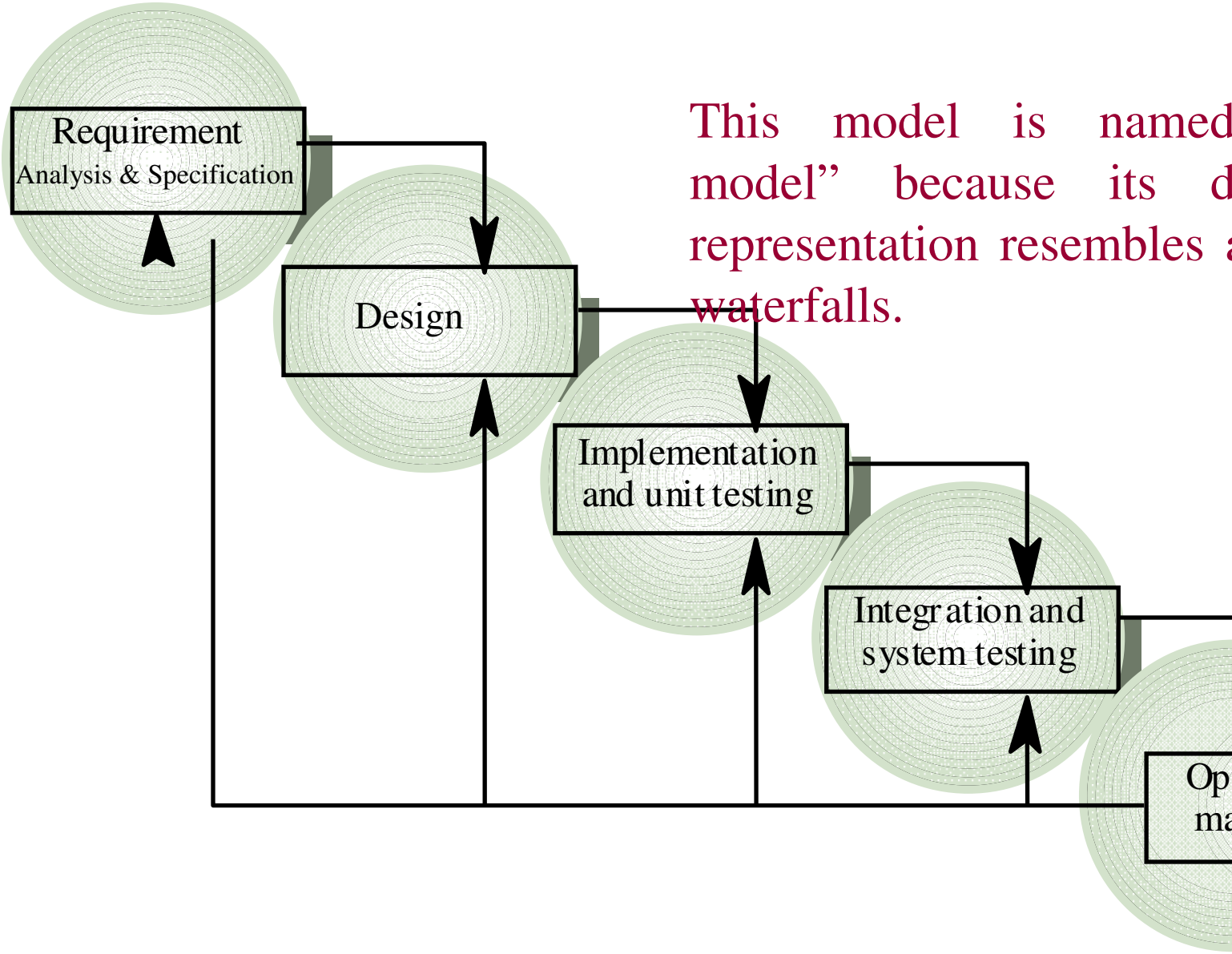
- Product is constructed without specifications or any attempt at design
- Adhoc approach and not well defined
- Simple two phase model



Build & Fix Model

- Suitable for small programming exercises of 100 or
- Unsatisfactory for software for any reasonable size
- Code soon becomes unfixable & unenhanceable
- No room for structured design
- Maintenance is practically not possible

Waterfall Model



This model is named model" because its d representation resembles a waterfalls.

Waterfall Model

This model is easy to understand and re
the notion of "define before design" and
before code".

The model expects complete &
requirements early in the process, v
unrealistic

Waterfall Model

Problems of waterfall model

i. It is difficult to define all requirements at the begi project

ii. This model is not suitable for accommodating any

iii. A working version of the system is not seen unt the project's life

iv. It does not scale up well to large projects.

v. Real projects are rarely sequential.

Incremental Process Models

They are effective in the situations where requiremen
defined precisely and there is no confusion abou
functionality of the final product.

After every cycle a useable product is given to the custom

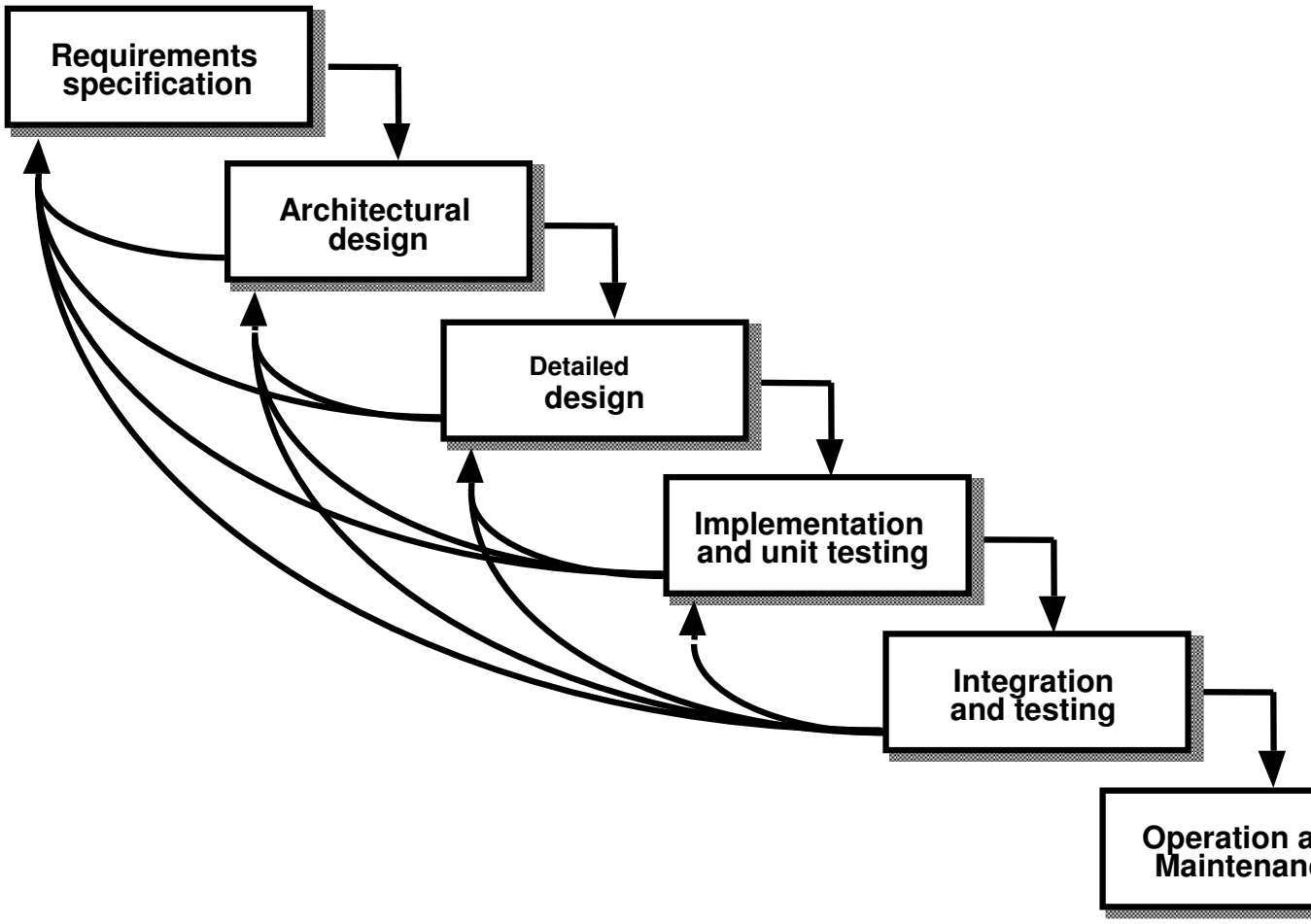
Popular particularly when we have to quickly deliver a li
functionality system.

Iterative Enhancement Mode

This model has the same phases as the waterfall mode
fewer restrictions. Generally the phases occur in the sa
in the waterfall model, but they may be conducted in sev
Useable product is released at the end of the each cycle
release providing additional functionality.

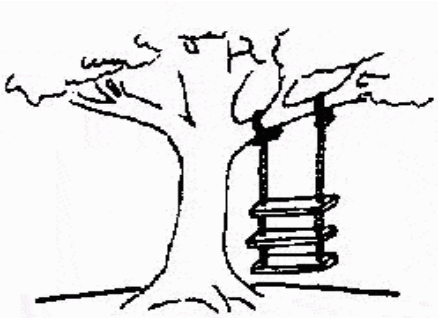
- ✓ Customers and developers specify as many requ
possible and prepare a SRS document.
- ✓ Developers and customers then prioritize these requ
- ✓ Developers implement the specified requirements
more cycles of design, implementation and test ba
defined priorities.

Iterative Enhancement Model

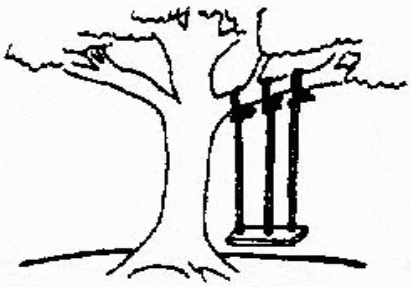


The Rapid Application Development (RAD

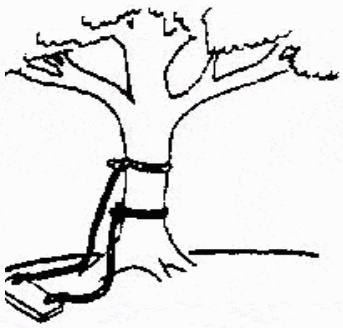
- Developed by IBM in 1980
- User participation is essential



The requirements specification was defined like this

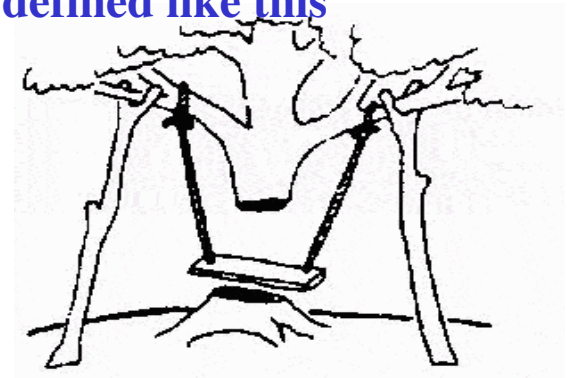


The developers understood it in that way

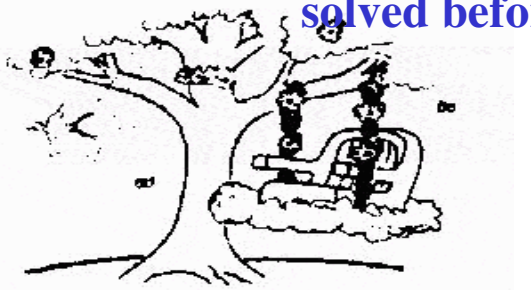


This is how the problem was solved before.

This is prob solve



That is the program after debugging



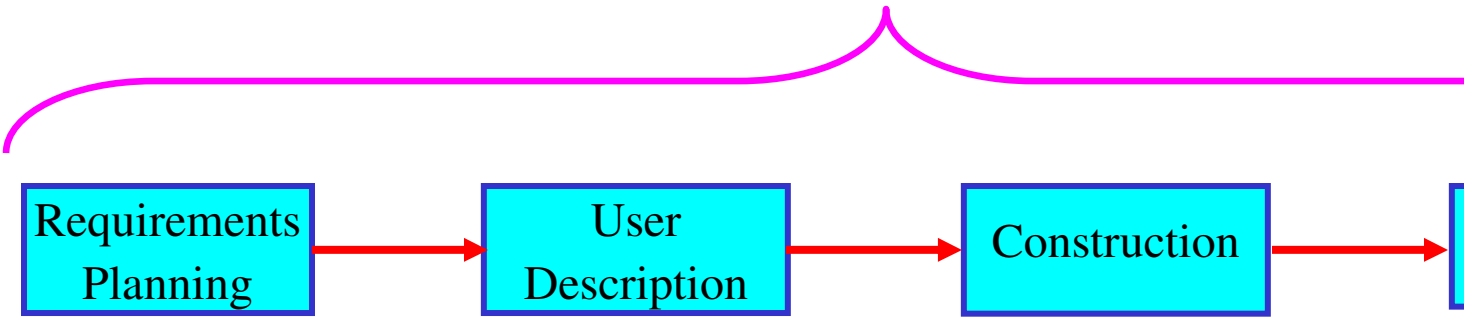
This is how the program is described by marketing department

This, in fact, customer wa

The Rapid Application Development (RAD

- Build a rapid prototype
- Give it to user for evaluation & obtain feedback
- Prototype is refined

With active participation of users



The Rapid Application Development (RAI

Not an appropriate model in the absence participation.

Reusable components are required to reduce dev time.

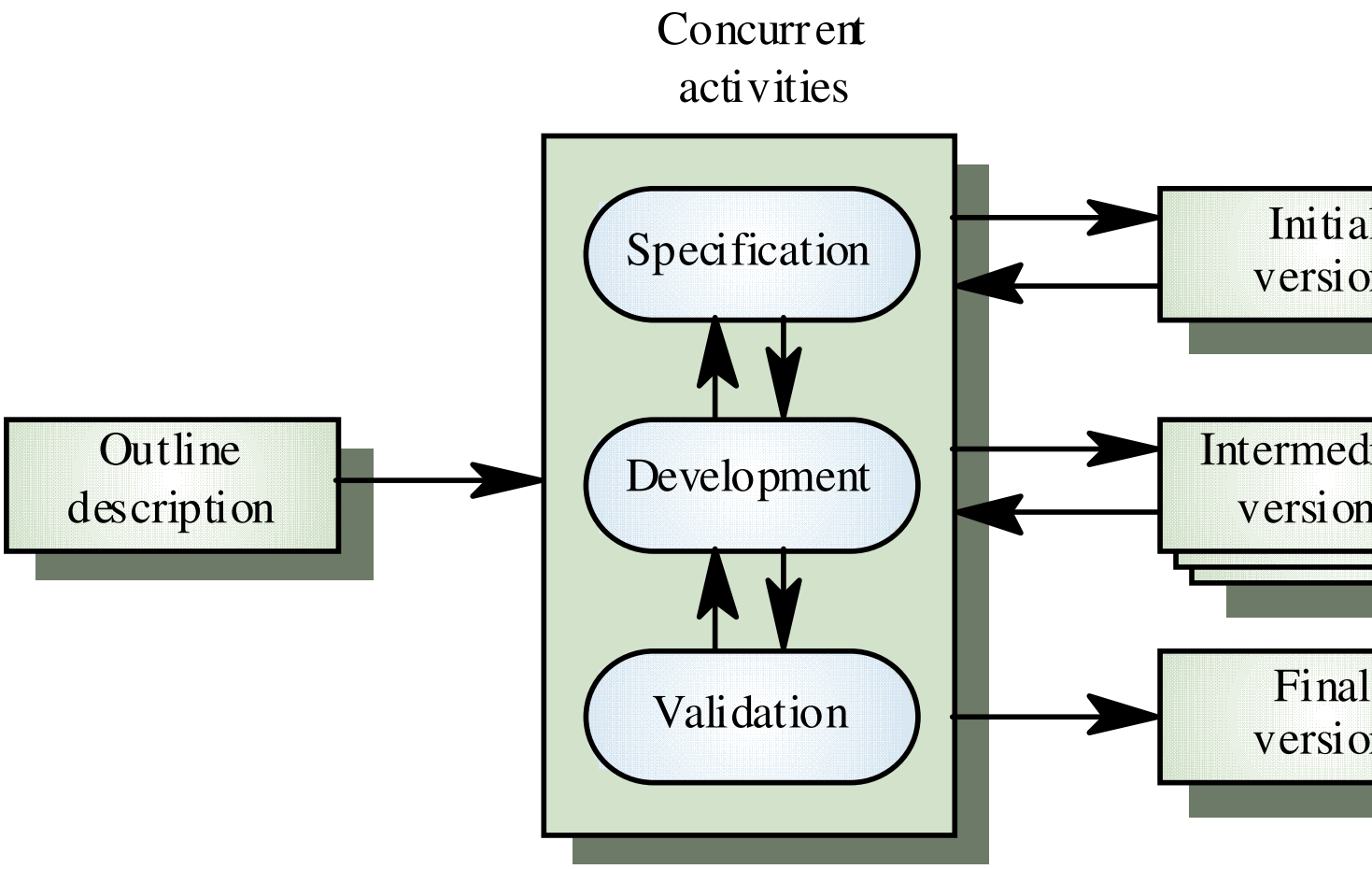
Highly specialized & skilled developers are req such developers are not easily available.

Evolutionary Process Models

Evolutionary process model resembles iterative en
model. The same phases as defined for the waterfall m
here in a cyclical fashion. This model differs fron
enhancement model in the sense that this does not
useable product at the end of each cycle. In ev
development, requirements are implemented by categ
than by priority.

This model is useful for projects using new technology
well understood. This is also used for complex project
functionality must be delivered at one time, but the re
are unstable or not well understood at the beginning.

Evolutionary Process Model

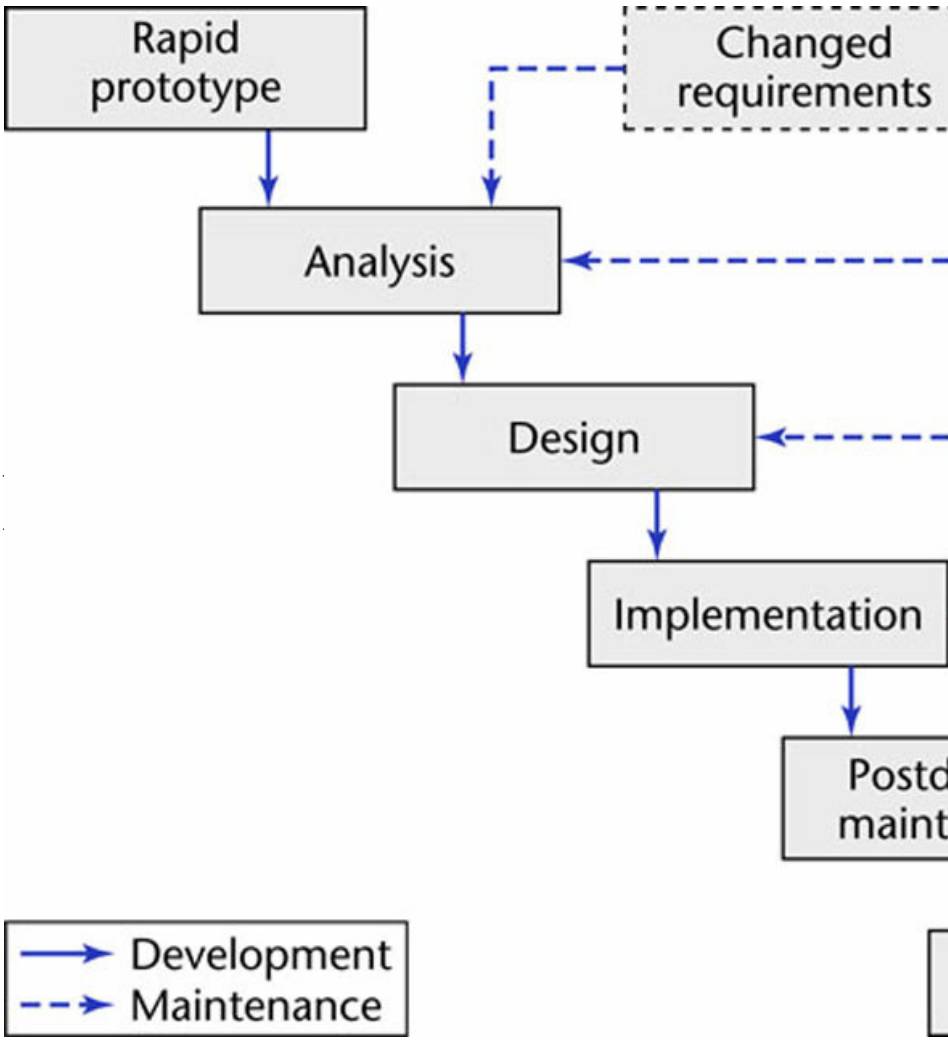


Prototyping Model

- The prototype may be a usable program but is not s
  the final software product.
- The code for the prototype is thrown away.
  experience gathered helps in developing the actual s
- The development of a prototype might involve extra
  overall cost might turnout to be lower than th
  equivalent system developed using the waterfall mo

Prototyping Model

- Linear mode
- "Rapid"



Spiral Model

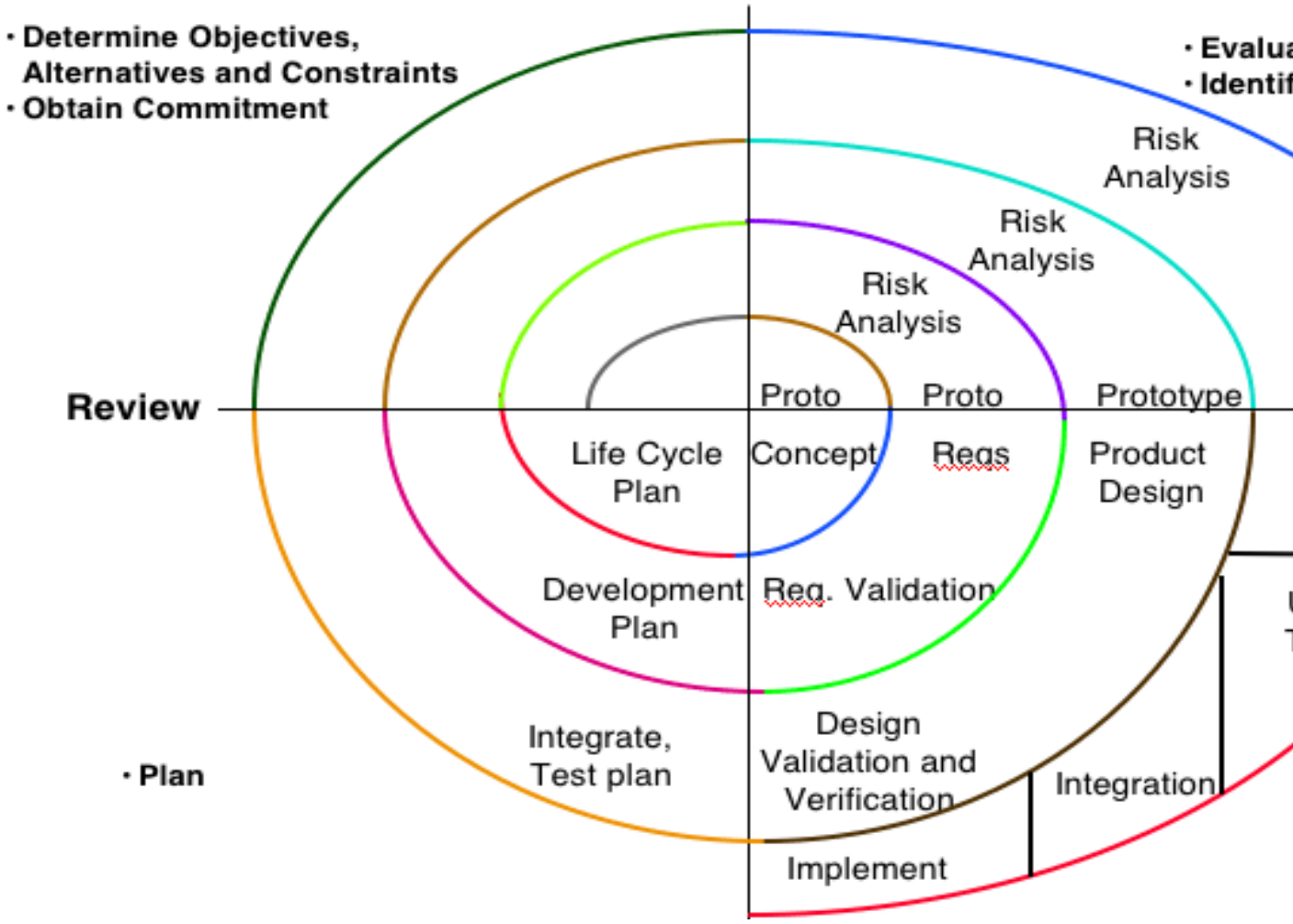
Models do not deal with uncertainly which is inherent t
projects.

Important software projects have failed because project
neglected & nobody was prepared when something
happened.

Barry Boehm recognized this and tired to incorporate t
risk" factor into a life cycle model.

The result is the spiral model, which was presented in 19

Spiral Model



Spiral Model

The radial dimension of the model represents the cumul
Each path around the spiral is indicative of increased
angular dimension represents the progress made in comp
cycle. Each loop of the spiral from X-axis clockwise th
represents one phase. One phase is split roughly into fou
major activities.

- **Planning:** Determination of objectives, alter
constraints.
- **Risk Analysis:** Analyze alternatives and attempts
and resolve the risks involved.
- **Development:** Product development and testing proc
- **Assessment:** Customer evaluation

Spiral Model

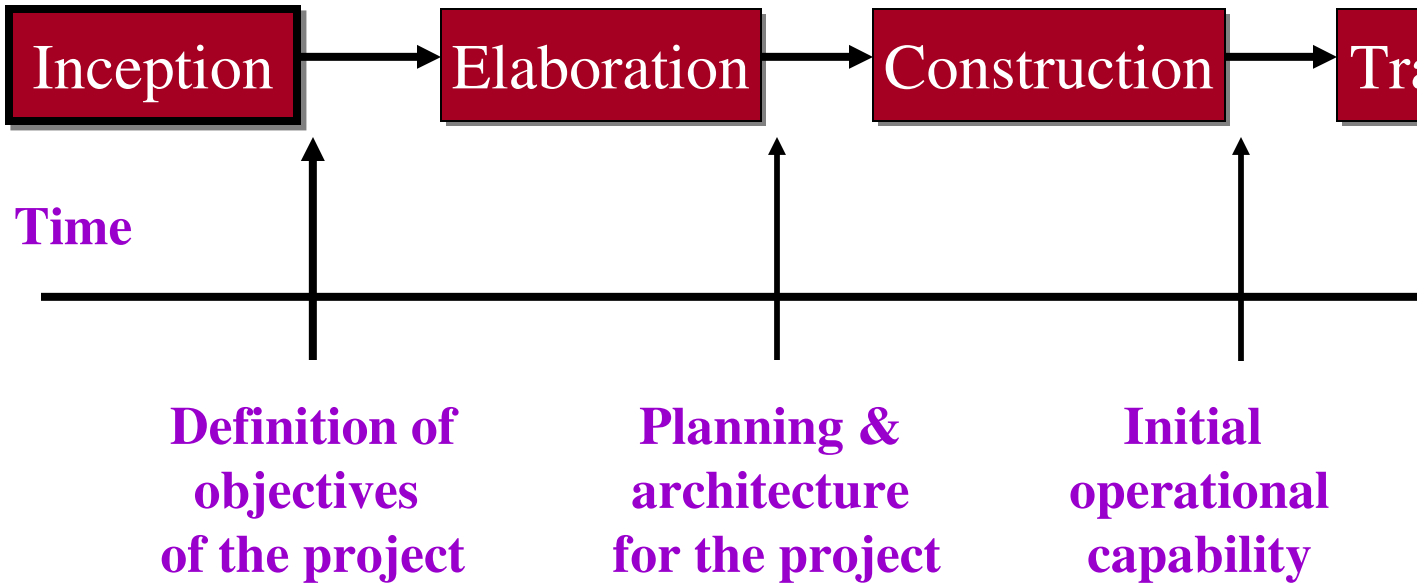
- An important feature of the spiral model is that ea completed with a review by the people concerne project (designers and programmers)
- The advantage of this model is the wide range of accommodate the good features of other life cycle m
- It becomes equivalent to another life cycle appropriate situations.

The spiral model has some difficulties that need to b before it can be a universally applied life cycle mo difficulties include lack of explicit process guidance in d objectives, constraints, alternatives; relying on risk expertise; and provides more flexibility than required applications.

The Unified Process

- Developed by I.Jacobson, G.Booch and J.Rumbaugh
- Software engineering process with the goal of prod quality maintainable software within specified time a
- Developed through a series of fixed length mini pro iterations.
- Maintained and enhanced by Rational Software Cor thus referred to as Rational Unified Process (RUP).

Phases of the Unified Process



Phases of the Unified Process

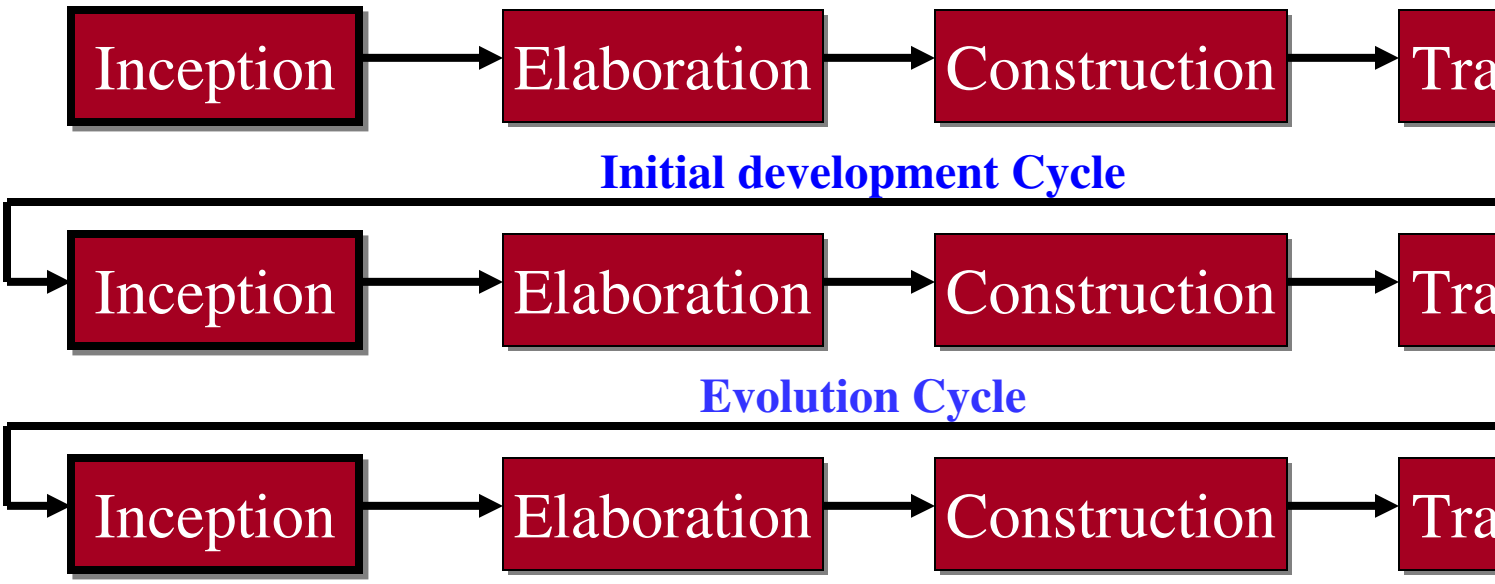
- Inception: defines scope of the project.
- Elaboration

  - How do we plan & design the project?

  - What resources are required?

  - What type of architecture may be suitable?
- Construction: the objectives are translated in architecture documents.
- Transition : involves many activities like deliverin supporting, and maintaining the product.

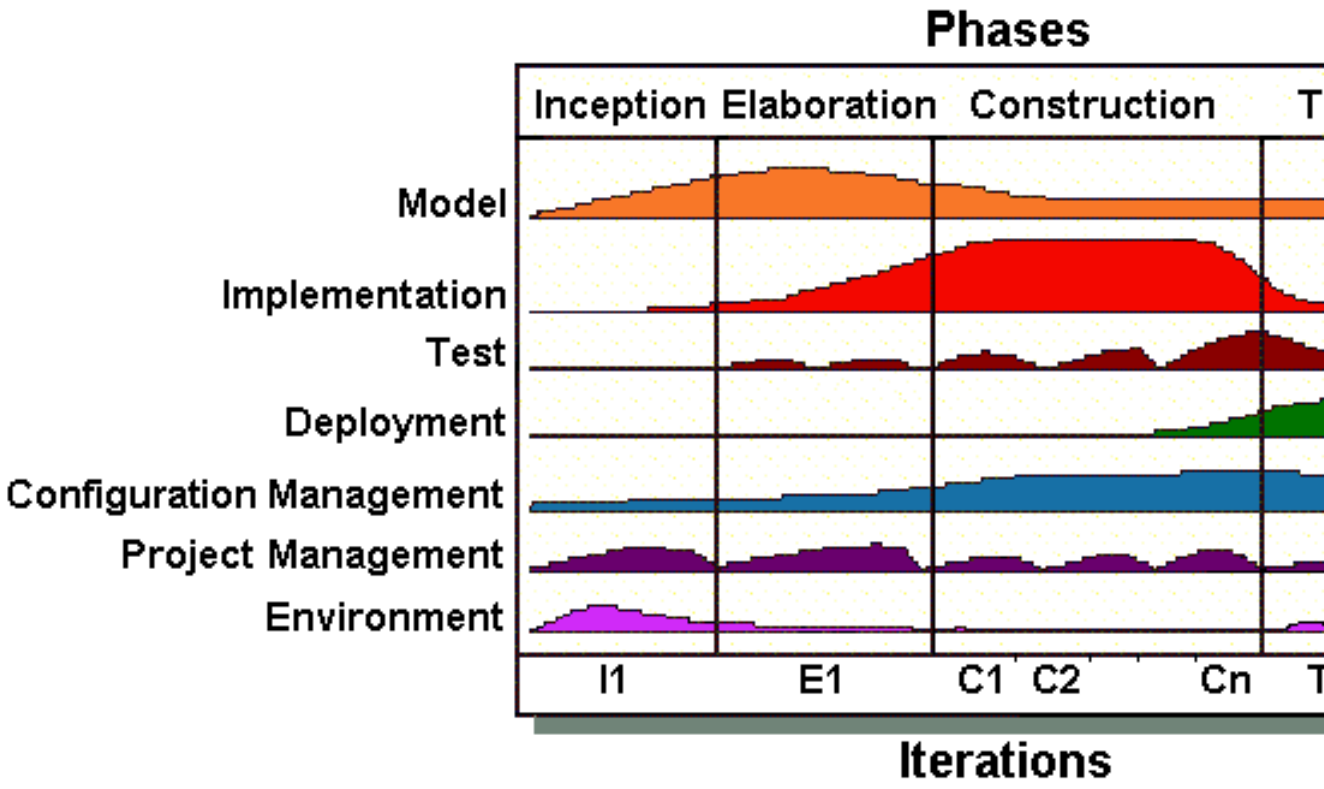
Initial development & Evolution Cyc



Continue till the product is reti

V1=version1, V2 =version2, V3=version3

Iterations & Workflow of Unified Pro

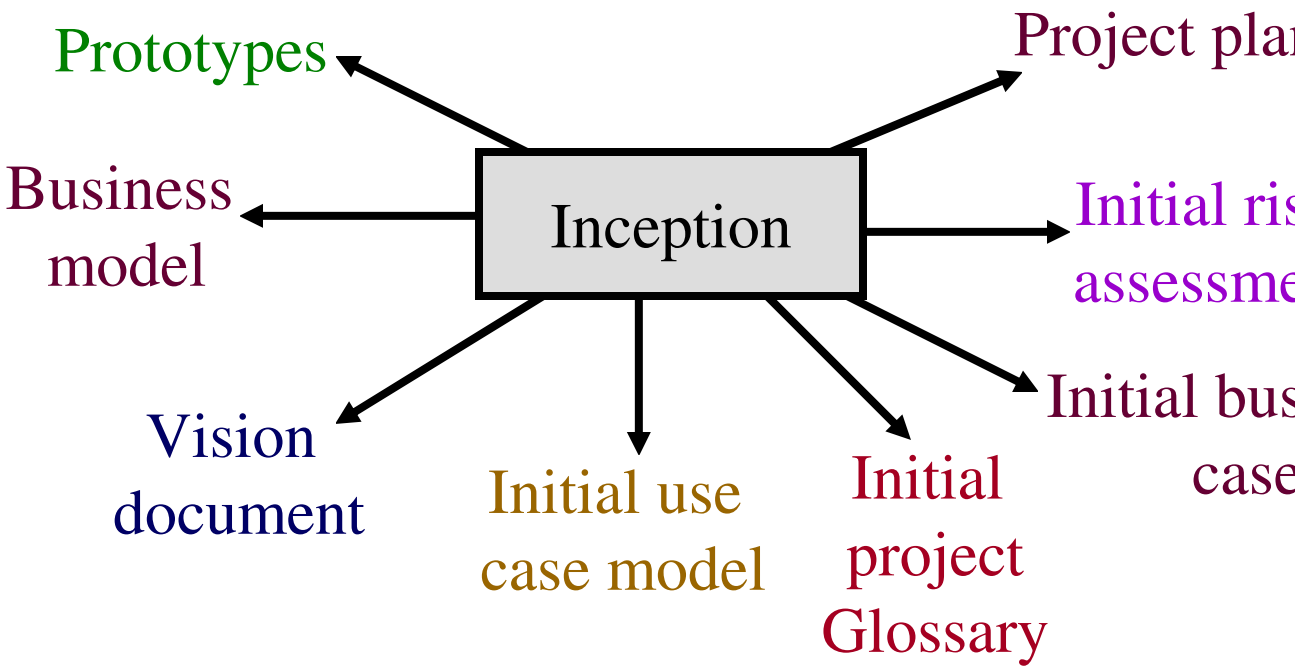


Inception Phase

The inception phase has the following objectives:

- Gathering and analyzing the requirements.
- Planning and preparing a business case and alternatives for risk management, scheduling reso
- Estimating the overall cost and schedule for the p
- Studying the feasibility and calculating profitab project.

Outcomes of Inception Phase

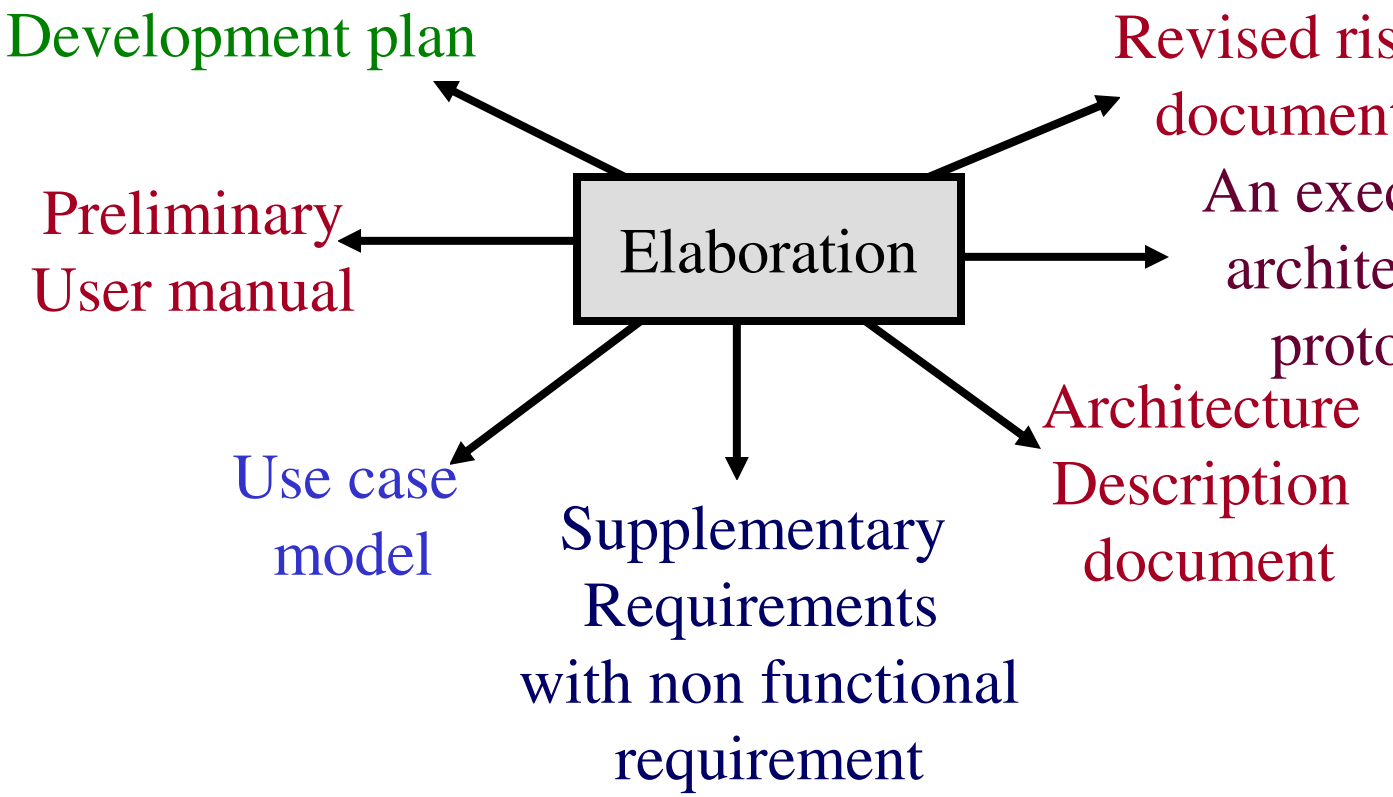


Elaboration Phase

The elaboration phase has the following objectives:

- Establishing architectural foundations.
- Design of use case model.
- Elaborating the process, infrastructure & d environment.
- Selecting component.
- Demonstrating that architecture support the reasonable cost & within specified time.

Outcomes of Elaboration Phase

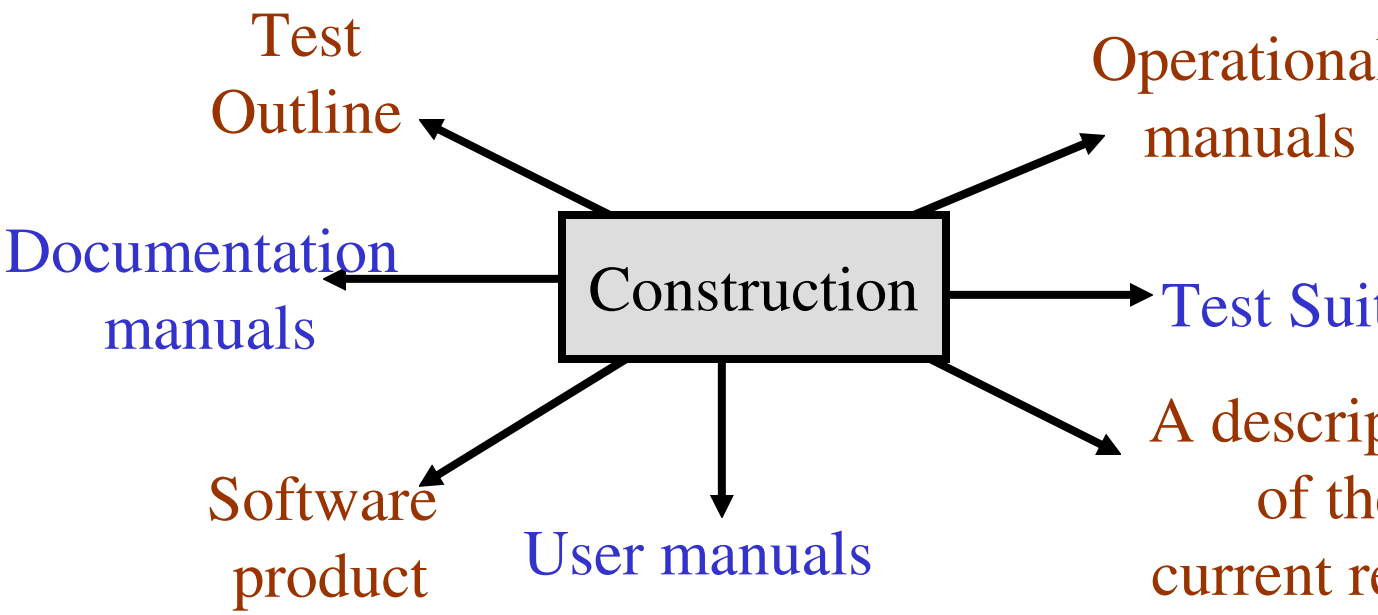


Construction Phase

The construction phase has the following objectives:

- Implementing the project.
- Minimizing development cost.
- Management and optimizing resources.
- Testing the product
- Assessing the product releases against acceptance

Outcomes of Construction Phase

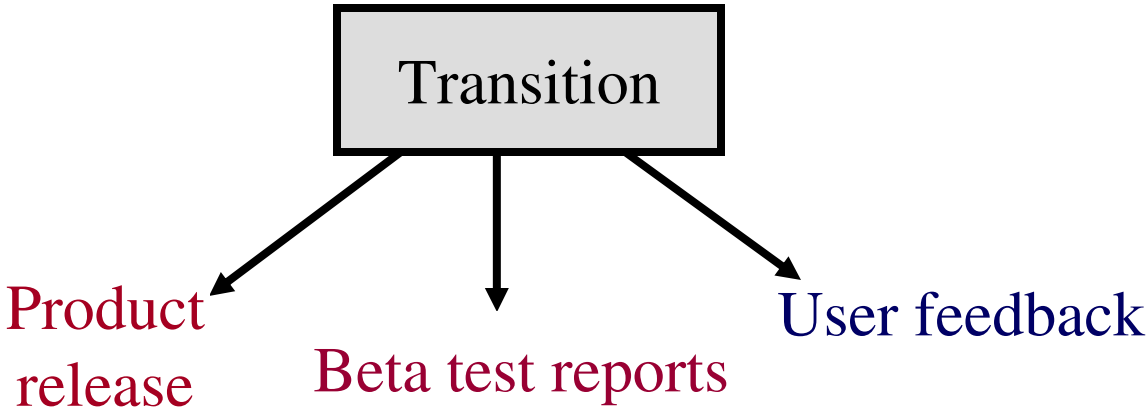


Transition Phase

The transition phase has the following objectives:

- Starting of beta testing
- Analysis of user's views.
- Training of users.
- Tuning activities including bug fixing and enhan performance & usability
- Assessing the customer satisfaction.

Outcomes of Transition Phase



Selection of a Life Cycle Mode

Selection of a model is based on:

a) Requirements

b) Development team

c) Users

d) Project type and associated risk

Based On Characteristics Of Requireme

Requirements	Waterfall	Prototype	Iterative enhancement	Evolutionary development	
Are requirements easily understandable and defined?	Yes	No	No	No	
Do we change requirements quite often?	No	Yes	No	No	
Can we define requirements early in the cycle?	Yes	No	Yes	Yes	
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes	

Based On Status Of Development T

Development team	Waterfall	Prototype	Iterative enhancement	Evolutionary development	S
Less experience on similar projects?	No	Yes	No	No	
Less domain knowledge (new to the technology)	Yes	No	Yes	Yes	
Less experience on tools to be used	Yes	No	No	No	
Availability of training if required	No	No	Yes	Yes	

Based On User's Participation

Involvement of Users	Waterfall	Prototype	Iterative enhancement	Evolutionary development	
User involvement in all phases	No	Yes	No	No	
Limited user participation	Yes	No	Yes	Yes	
User have no previous experience of participation in similar projects	No	Yes	Yes	Yes	
Users are experts of problem domain	No	Yes	Yes	Yes	

Based On Type Of Project With Associate

Project type and risk	Waterfall	Prototype	Iterative enhancement	Evolutionary development	
Project is the enhancement of the existing system	No	No	Yes	Yes	
Funding is stable for the project	Yes	Yes	No	No	
High reliability requirements	No	No	Yes	Yes	
Tight project schedule	No	Yes	Yes	Yes	
Use of reusable components	No	Yes	No	No	
Are resources (time, money, people etc.) scare?	No	Yes	No	No	

Multiple Choice Questions

Note: Select most appropriate answer of the following question

2.1 Spiral Model was developed by

(a) Bev Littlewood (b) Berry Boehm

(c) Roger Pressman (d) Victor Basili

2.2 Which model is most popular for student's small projects?

(a) Waterfall model (b) Spiral model

(c) Quick and fix model (d) Prototyping model

2.3 Which is not a software life cycle model?

(a) Waterfall model (b) Spiral model

(c) Prototyping model (d) Capability maturity m

2.4 Project risk factor is considered in

(a) Waterfall model (b) Prototyping model

(c) Spiral model (d) Iterative enhancemen

2.5 SDLC stands for

(a) Software design life cycle (b) Software developmer

(c) System development life cycle (d) System design life cy

Multiple Choice Questions

Note: Select most appropriate answer of the following question

2.6 Build and fix model has

(a) 3 phases (b) 1 phase

(c) 2 phases (d) 4 phases

2.7 SRS stands for

(a) Software requirements specification (b) Software requirement

(c) System requirements specification (d) none of the above

2.8 Waterfall model is not suitable for

(a) small projects (b) accommodating chan

(c) complex projects (d) none of the above

2.9 RAD stands for

(a) Rapid application development (b) Relative application d

(c) Ready application development (d) Repeated application

2.10 RAD model was proposed by

(a) Lucent Technologies (b) Motorola

(c) IBM (d) Microsoft

Multiple Choice Questions

Note: Select most appropriate answer of the following question

2.11 If requirements are easily understandable and defined,which model

(a) Waterfall model
(b) Prototyping model
(c) Spiral model
(d) None of the above

2.12 If requirements are frequently changing, which model is to be select

(a) Waterfall model
(b) Prototyping model
(c) RAD model
(d) Iterative enhancemen

2.13 If user participation is available, which model is to be chosen?

(a) Waterfall model
(b) Iterative enhancemen
(c) Spiral model
(d) RAD model

2.14 If limited user participation is available, which model is to be selecte

(a) Waterfall model
(b) Spiral model
(c) Iterative enhancement model
(d) any of the above

2.15 If project is the enhancement of existing system, which model is best

(a) Waterfall model
(b) Prototyping model
(c) Iterative enhancement model
(d) Spiral model

Multiple Choice Questions

Note: Select most appropriate answer of the following question

2.16 Which one is the most important feature of spiral model?

(a) Quality management (b) Risk management

(c) Performance management (d) Efficiency manageme

2.17 Most suitable model for new technology that is not well understood

(a) Waterfall model (b) RAD model

(c) Iterative enhancement model (d) Evolutionary develop

2.18 Statistically, the maximum percentage of errors belong to the follow SDLC

(a) Coding (b) Design

(c) Specifications (d) Installation and maint

2.19 Which phase is not available in software life cycle?

(a) Coding (b) Testing

(c) Maintenance (d) Abstraction

2.20 The development is supposed to proceed linearly through the phase i

(a) Spiral model (b) Waterfall model

(c) Prototyping model (d) None of the above

Multiple Choice Questions

Note: Select most appropriate answer of the following question

2.21 Unified process is maintained by

(a) Infosys (b) Rational software cor

(c) SUN Microsystems (d) None of the above

2.22 Unified process is

(a) Iterative (b) Incremental

(c) Evolutionary (d) All of the above

2.23 Who is not in the team of Unified process development?

(a) I.Jacobson (b) G.Booch

(c) B.Boehm (d) J.Rumbaugh

2.24 How many phases are in the unified process?

(a) 4 (b) 5

(c) 2 (d) None of the above

2.25 The outcome of construction phased can be treated as:

(a) Product release (b) Beta release

(c) Alpha release (d) All of the above

Exercises

2.1 What do you understand by the term Software Developme (SDLC)? Why is it important to adhere to a life cycle developing a large software product?

2.2 What is software life cycle? Discuss the generic waterfall m

2.3 List the advantages of using waterfall model instead of ad fix model.

2.4 Discuss the prototyping model. What is the effect of prototype on the overall cost of the project?

2.5 What are the advantages of developing the prototype of a sy

2.6 Describe the type of situations where iterative enhancemen lead to difficulties.

2.7 Compare iterative enhancement model and evolutionary pro

Exercises

2.8 Sketch a neat diagram of spiral model of software life cycle

2.9 Compare the waterfall model and the spiral model development.

2.10 As we move outward along with process flow path of the what can we say about software that is being developed or main

2.11 How does "project risk" factor effect the spiral model development.

2.12 List the advantages and disadvantages of involving a soft throughout the software development planning process.

2.13 Explain the spiral model of software development. limitations of such a model?

2.14 Describe the rapid application development (RAD) model phase in detail.

2.15 What are the characteristics to be considered for the select cycle model?

Exercises

2.16 What is the role of user participation in the selection c model?.

2.17 Why do we feel that characteristics of requirements significant role in the selection of a life cycle model?

2.18 Write short note on "status of development team" for the life cycle model?.

2.19 Discuss the selection process parameters for a life cycle m

2.20 What is unified process? Explain various phases along wit of each phase.

2.21 Describe the unified process work products after each ph process.

2.22 What are the advantages of iterative approach over sequen Why is unified process called as iterative or incremental?