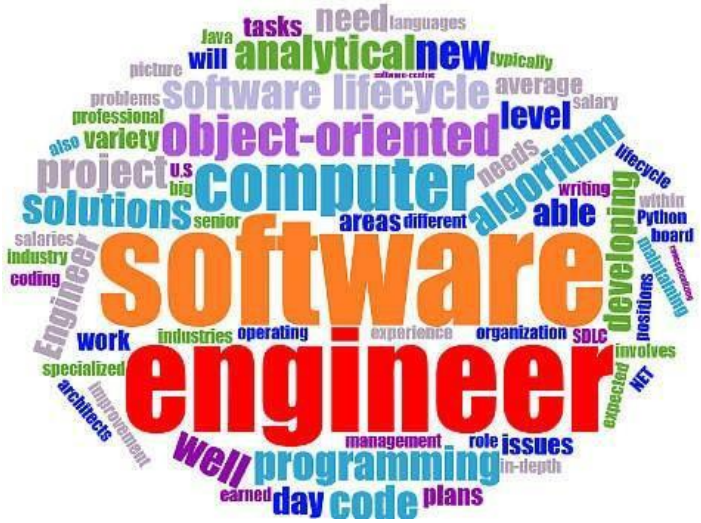
THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

UCS503
Software Engineering



Slide Set - 1

Organized & Presented By:
Software Engineering Team CSED
TIET, Patiala

Course Syllabus

UCS503 SOFTWARE ENGINEERING **L: T: P :Cr**

3:0:2:4

Software Engineering and Processes: Introduction to Software Engineering, Software Evolution, Software Characteristics, Software Crisis: Problem and Causes, Software process models (Waterfall, Incremental, and Evolutionary process models and Agile), Software quality concepts, process improvement, software process capability maturity models, Personal Software process and Team Software Process, Overview of Agile Process.

Requirements Engineering: Problem Analysis, Requirement elicitation and Validation, Requirements modeling: Scenarios, Information and analysis classes, flow and behavioral modeling, documenting Software Requirement Specification (SRS).

Software Design and construction: System design principles: levels of abstraction (architectural and detailed design), separation of concerns, information hiding, coupling and cohesion, Structured design (top-down functional decomposition), object-oriented design, event driven design, component-level design, test driven design, data-structured centered, aspect oriented design , function oriented, service oriented, Design patterns, Coding Practices: Techniques, Refactoring, Integration Strategies, Internal Documentation.

Software Verification and Validation: Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Specification, Software Testing Strategies, Verification & Validation, Unit, Integration Testing, Top Down and Bottom Up Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Debugging.

Software Project Management: SP Estimation of scope (LOC, FP etc), time (PERT/CPM Networks), and cost (COCOMO models), Quality Management, Plan for software Quality Control and Assurance, Earned Value Analysis.

Advanced Topics: Formal specification, CASE Tools, Software Business Process Reengineering, Configuration Management.

Text Books (author, title, publisher and year)

- ***Text Books***
  - *Pressman S. R. and Maxim R. B., Software Engineering, A Practitioner's Approach, McGraw Hill International (2015) 7th Edition.*
  - *Sommerville I., Software Engineering, Addison-Wesley Publishing Company (2011) 9th Edition.*

- ***Reference Book***
  - *Foster C. E., Software Engineering: A Methodical Approach, Apress (2014)1st ed.*
  - *Booch G., Rambaugh J., Jacobson I., The Unified Modeling Language User Guide (2005) 2nd Edition.*

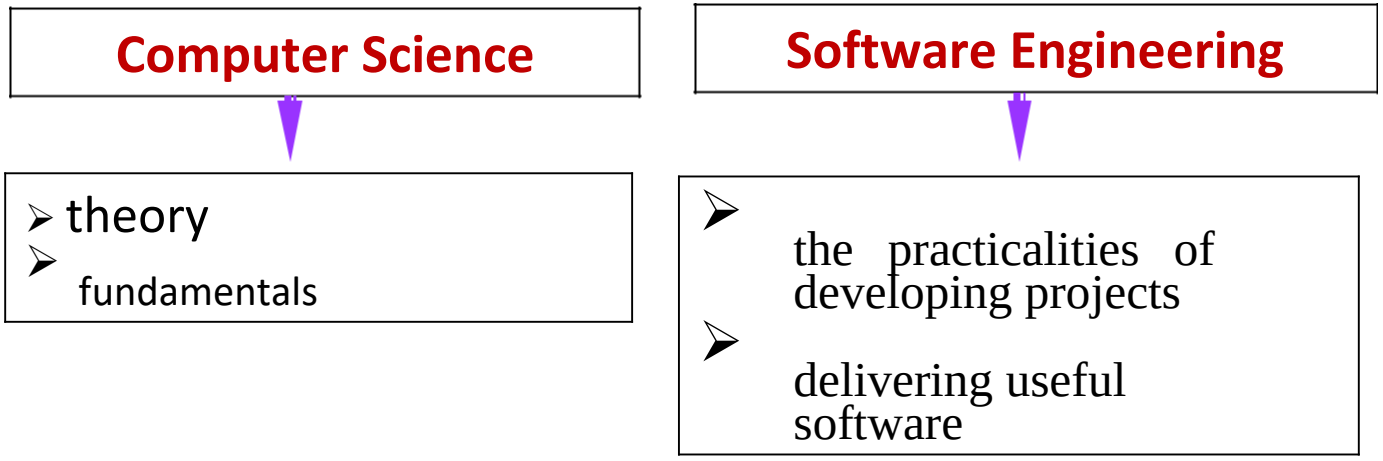
Course Learning Outcomes

- Analyze software development process models, including agile models and traditional models like waterfall.
- Demonstrate the use of software life cycle through requirements gathering, choice of process model and design model.
- Apply and use various UML models for software analysis, design and testing.
- Acquire knowledge about the concepts of application of formal specification, case tools and configuration management for software development.
- Analysis of software estimation techniques for creating project baselines.

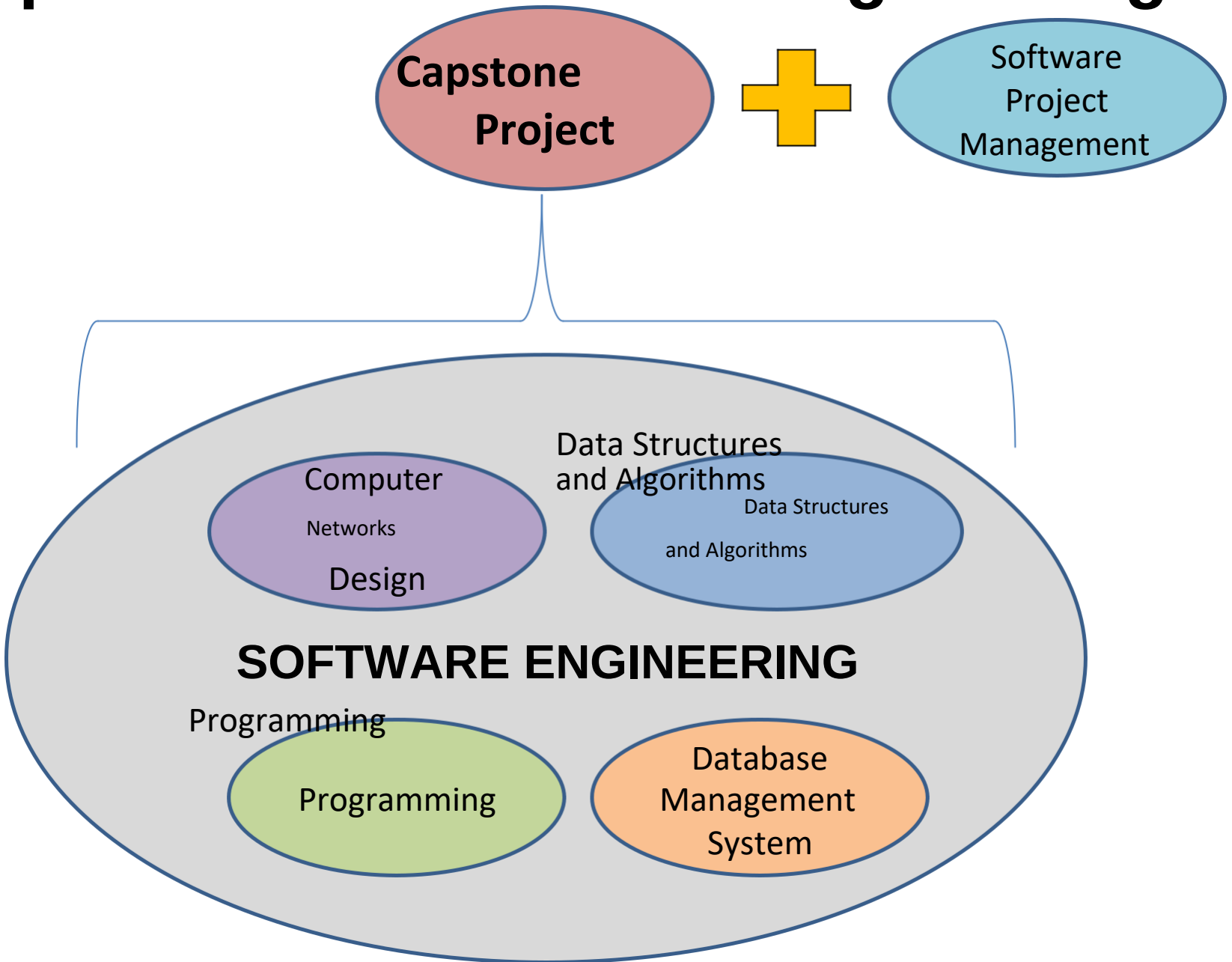
Lecture Objectives

- Identify the scope and necessity of software engineering.
- Identify the causes and solutions for software crisis.
- Discussion on Software Myths.
- Differentiate a program from a software product.

Introduction to Software Engineering

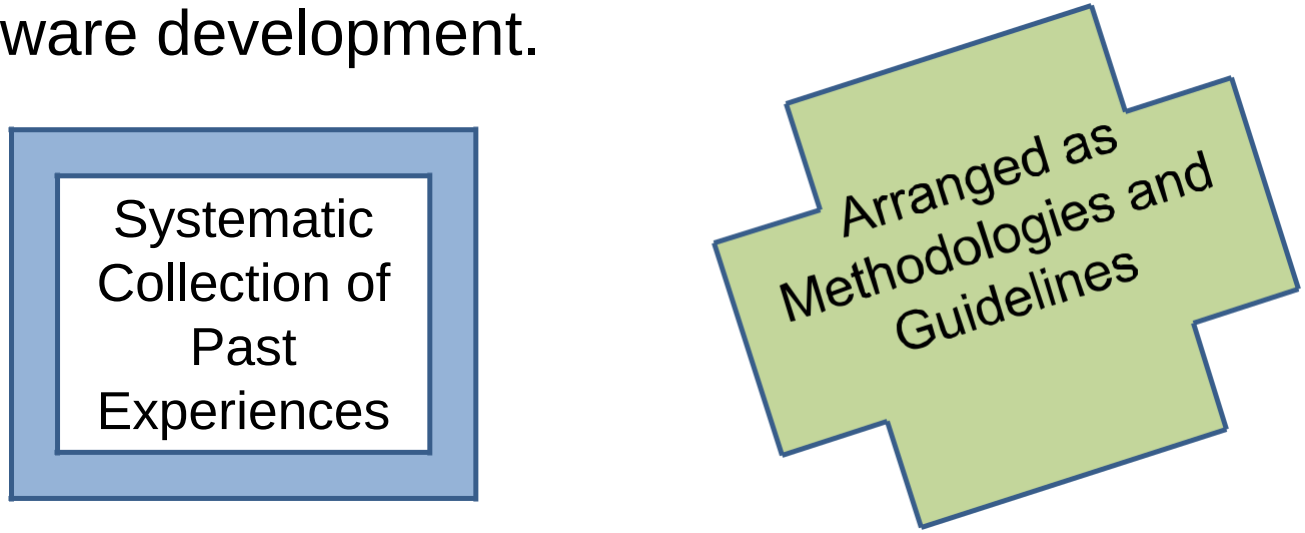


Importance of Software Engineering



Scope and Necessity of Software Engineering

- Software engineering is an engineering approach for software development.



- A small program can be written without using software engineering principles.

- But if one wants to develop a large software product, then software engineering principles are indispensable to achieve a good quality software cost effectively.

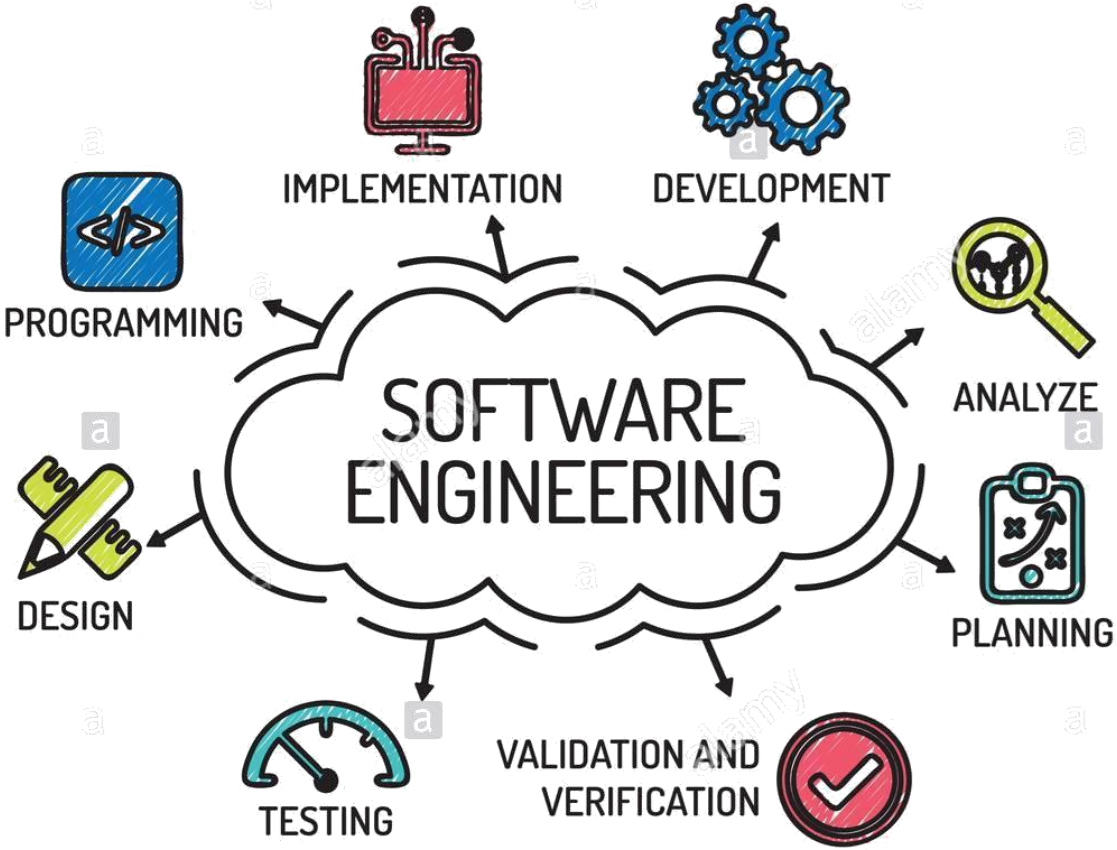
Example: Building a wall

- *Can be built with minimum knowledge*
- *Much expertise not required*

Constructing a Multi-storeyed Building

requisite knowledge about

- ✓ the strength of materials,
- ✓ testing,
- ✓ planning,
- ✓ architectural design



Scope and Necessity of Software Engineering

- Without using software engineering principles, it would be difficult to develop large programs. In industry, it is usually needed to develop large programs to accommodate multiple functions.

- A problem with developing such **large commercial programs** is the **complexity and difficulty levels of the programs increase exponentially with their sizes** as shown in Fig. 1.1

Contd...

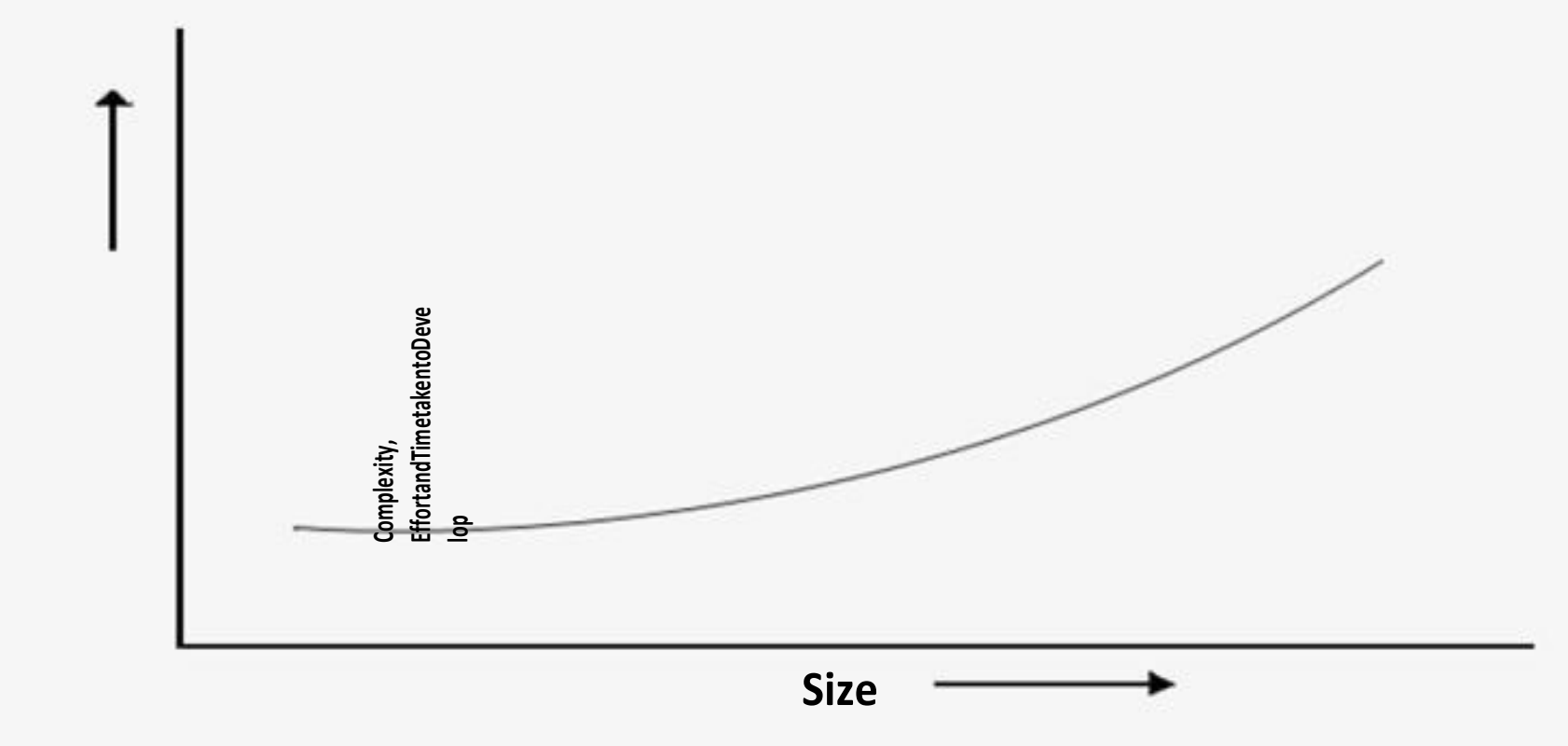


Fig. 1.1: Increase in development time and effort with problem size

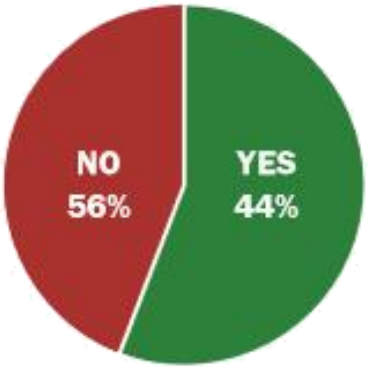
Scope and Necessity of Software Engineering…

- For example , a program of size 1,000 Lines of Code (LOC) has some complexity.
- So**, a program with 10,000 LOC is not just 10 times more difficult to develop**, but may as well turn out to be 100 times more difficult unless software engineering principles are used.

- In such situations software engineering techniques come to rescue.
- **Software engineering helps to reduce the programming complexity.** Software engineering principles use two important techniques to reduce problem complexity:

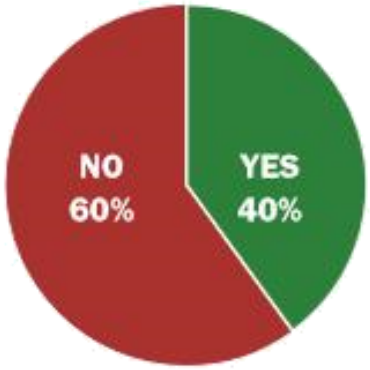
  abstraction and decomposition.

ONBUDGET



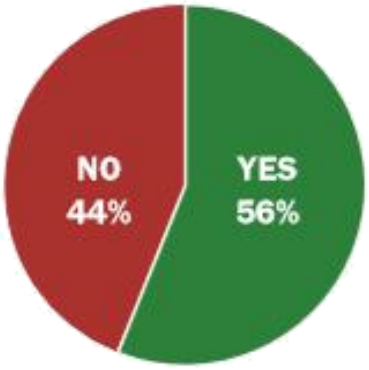
The percentage of projects that were OnBudget from FY2011–2015 within the new CHAOS database.

ONTIME



The percentage of projects that were OnTime from FY2011–2015 within the new CHAOS database.

ONTARGET



The percentage of projects that were OnTarget from FY2011–2015 within the new CHAOS database.

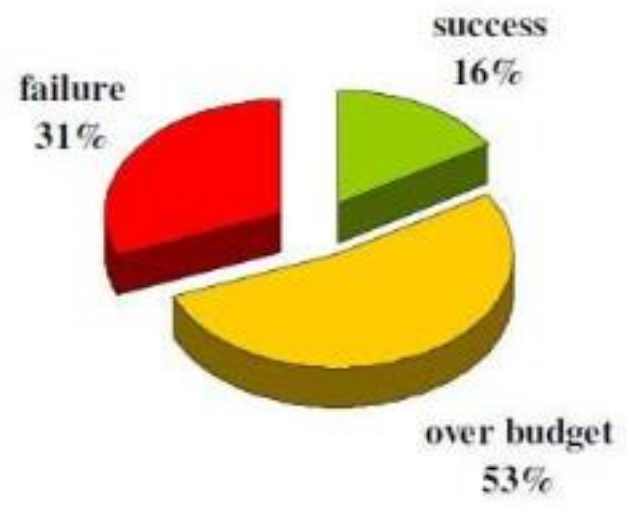
TRADITIONAL RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

The Traditional resolution of all software projects from FY2011–2015 within the new CHAOS database.

Evolving Role of Software

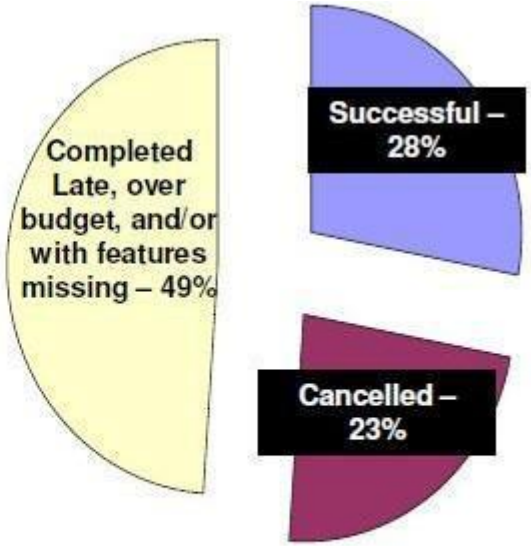
- Software industry is in Crisis!



Source: The Standish Group International, Inc. (CHAOS research)

Evolving Role of Software…

This is the SORRY state of Software Engineering Today!



- Data on 28,000 projects completed in 2000

IBM Data Compilation (2001)

As per the IBM report, "31%of the project get cancelled before they are completed, 53% over-run their cost estimates by an average of 189% and for every 100 projects, there are 94 restarts".

Evolving Role of Software…

Managers and Technical Persons are asked:

Why does it take so long to get the program finished?

Why cannot we find all errors before release?

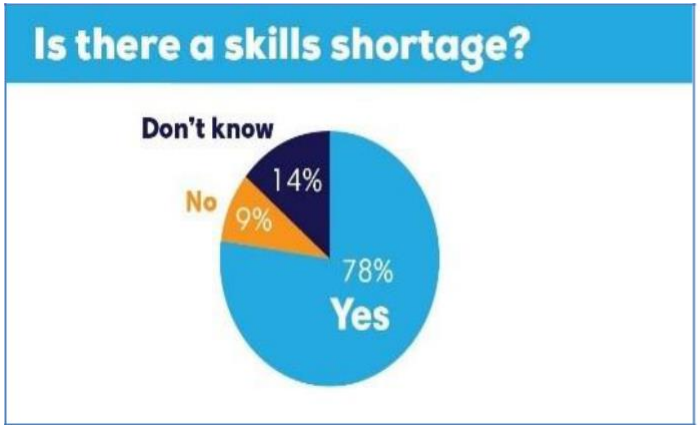
Why are costs so high?

Why do we have difficulty in measuring progress of software development ?

Factors Contributing to Software Crisis

Large Problems

Lack of adequate training in software



Increasing skill shortage

Low productivity improvements

Some Software Failures

https://www.youtube.com/watch?v=EMVBLg2MrLs

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…but the programmers decided to see if they could come up with a systematic way to eliminate these generic sorts of bugs in the future. A random group of programmers applied this system to the fuel dump module and other modules.

Seventeen additional, previously unknown problems surfaced!

Some Software Failures…

Financial Software

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

Some Software Failures…

Windows XP

- Microsoft released Windows XP on October 25, 2001.
- On the same day company posted 18 MB of compatibility patches on the website for bug fixes, compatibility updates, and enhancements.
- Two patches fixed important security holes.

The Worst Computer Bugs in History: Race conditions in Therac-25

https://www.youtube.com/watch?v=uEvu2PlDhO0

What is Software?

The product that software professionals **build** and then **support** over the long term.

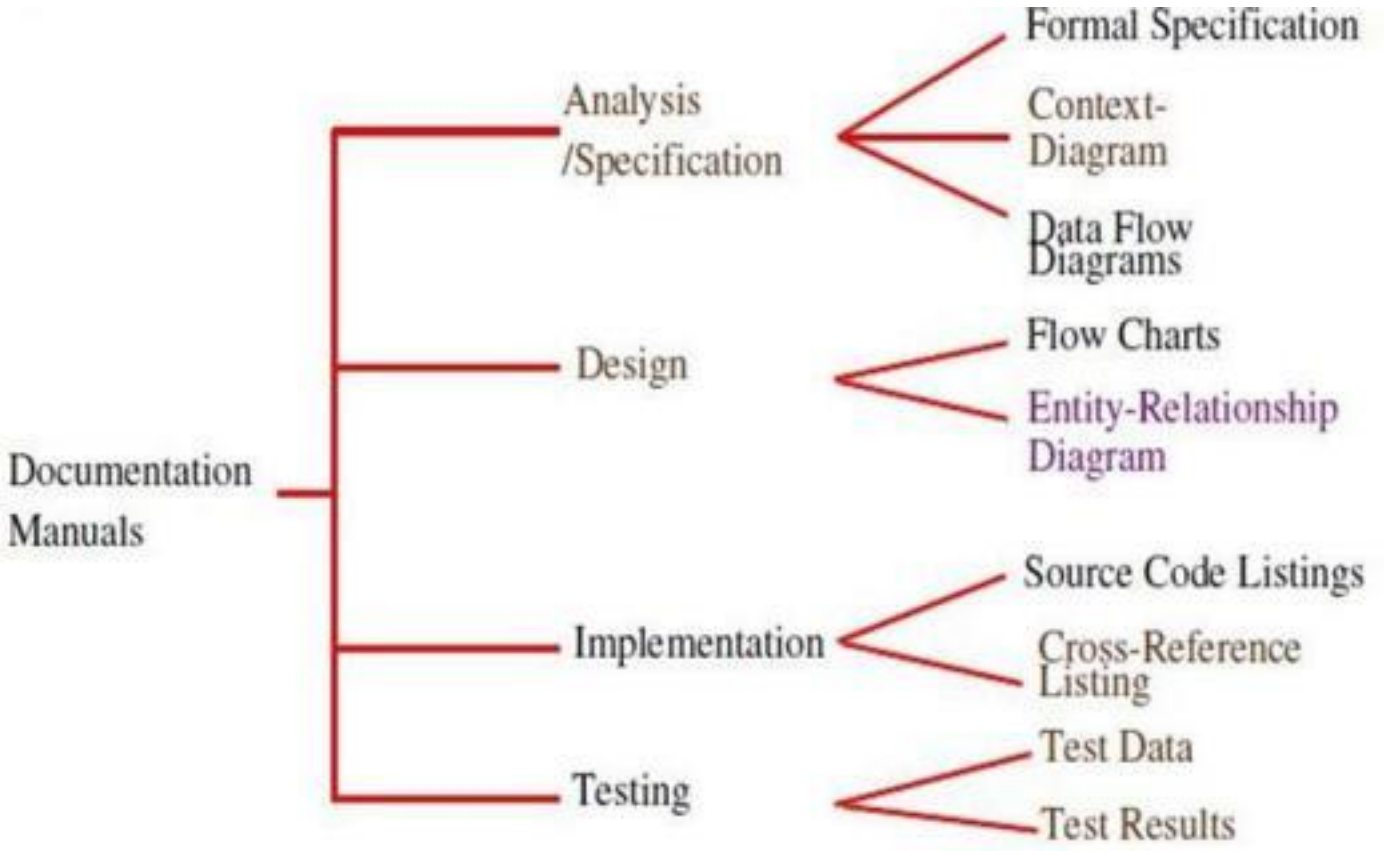
Software encompasses:

(1) Instructions (computer programs) that when executed provide desired features, function and performance.

(2) Data Structures that enable the programs to adequately store and manipulate information and

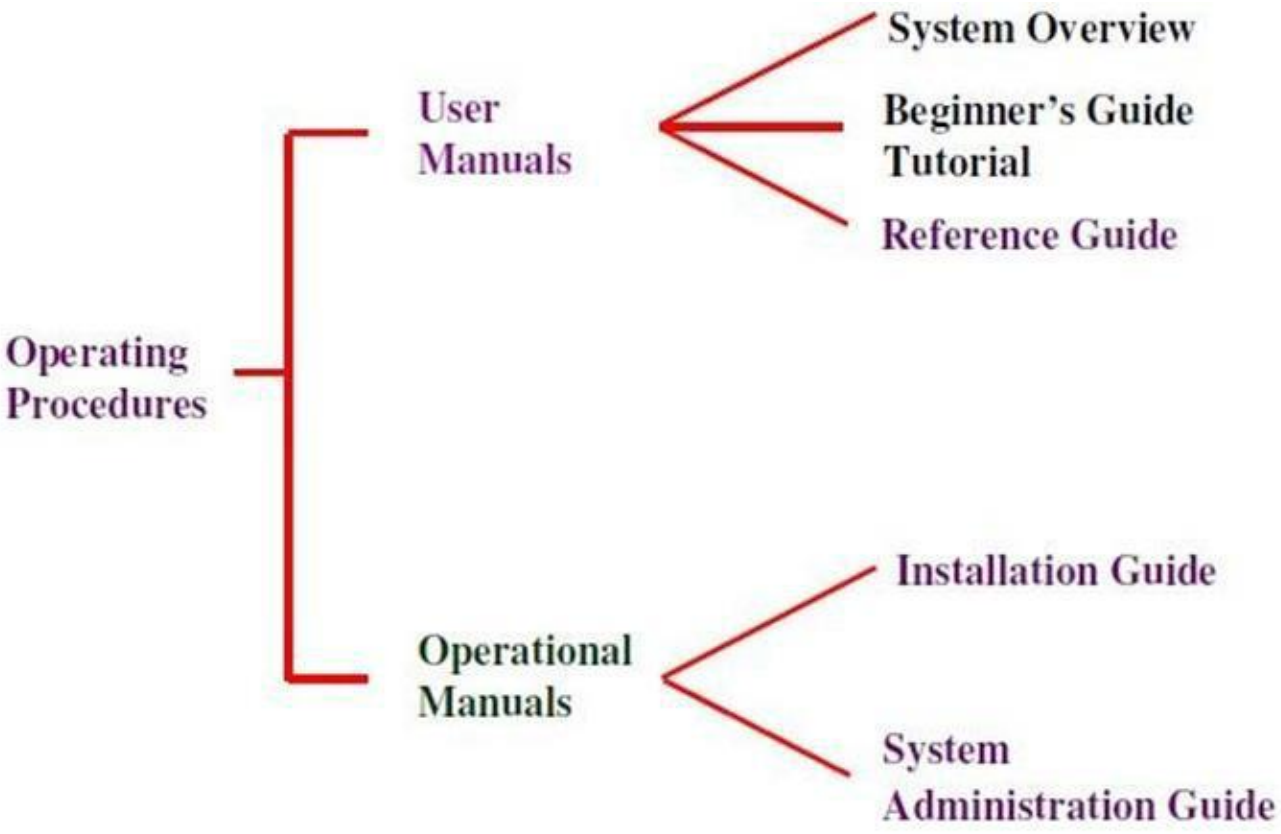
(3) Documentation that describes the operation and use of the programs.

Documentation consists of different types of manuals are



List of documentation manuals

Documentation consists of different types of manuals are

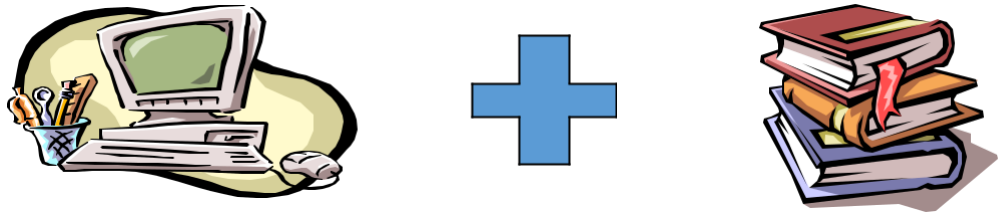


List of operating procedure manuals.

Software Products

- **Software product is defined as :**

  Computer programs and **associated documentation**



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

Types of Software Products

Generic

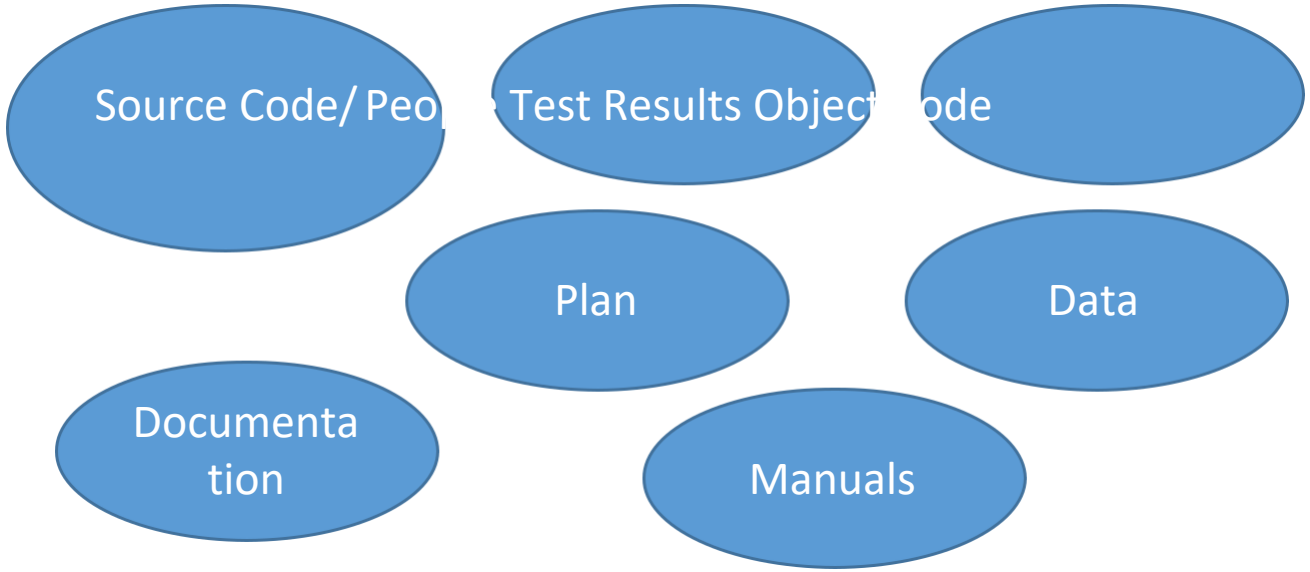
- Developed to be sold to a range of different customers
- Sold on the open market to any customer who is able to buy them.
- For Example:
  - Word processor – Word.
  - Spreadsheet – Excel.

Bespoke/Custom/ tailor-made software

- Developed for a single customer or organization according to their specification
- Sold to a specific customer.
- For Example:
  - software for the cafe franchise
- Also called as COTS **(Commercial Off The Shelf)**

Software Products

- Software product is a product designated for the delivery to the user.



Software Process

- A **set of activities** whose goal is the development or evolution of software
- Generic activities in all software processes are:
  - **Specification** - what the system should do and its development constraints
  - **Development** - production of the software system
  - **Validation** - checking that the software is what the customer wants
  - **Evolution** - changing the software in response to changing demands

Why its hard to improve software process

- **Lack of knowledge**

  Several software developers aren't aware of best practices of industry. In fact best practices obtainable in literature aren't being used widespread in software development.

- **Not enough time**

  There is forever a shortage of time because upper management are always demanding more software of higher quality in minimum possible time. Unrealistic schedule occasionally leave insufficient time to do the essential project work.

Cont…

CMM (Capability Maturity Model)

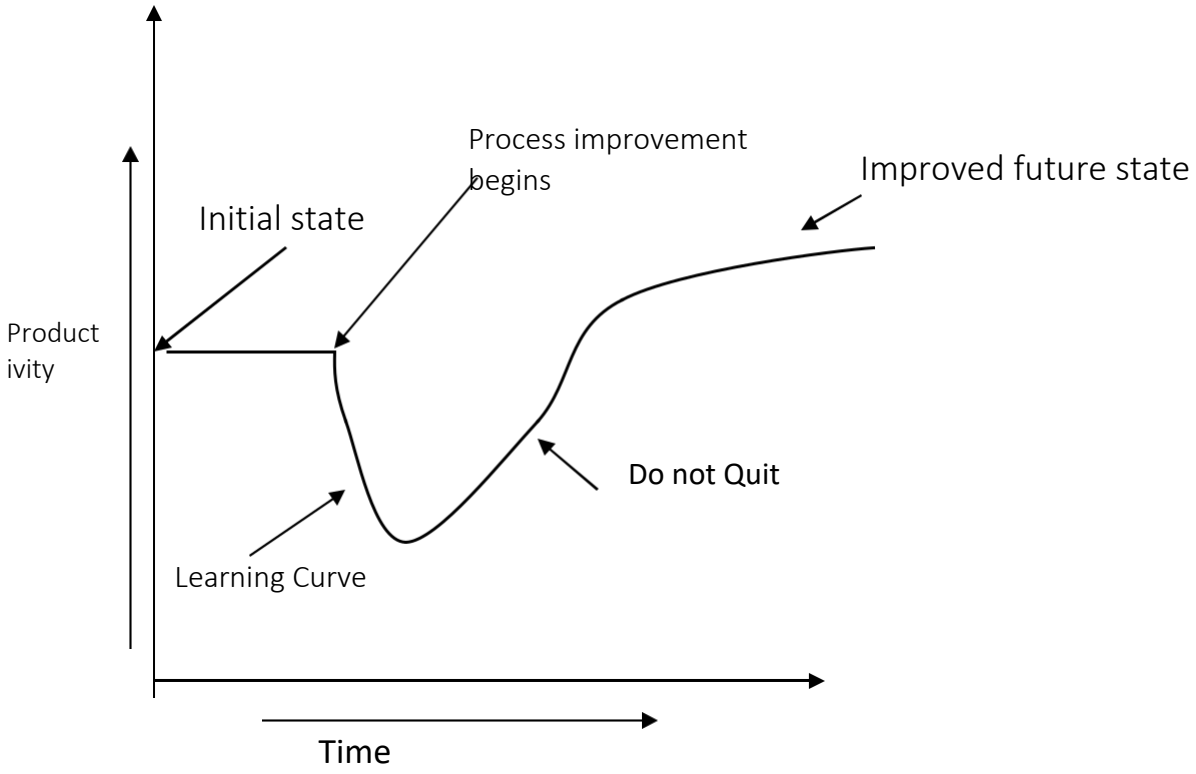
- **Wrong motivations**

  The process enhancement initiatives are taken for wrong reasons like sometimes contractor is demanding achievement of CMM or occasionally senior management is directing the organization to achieve CMM without a clear explanations why improvement was needed and its benefits.

- **Insufficient commitments**

  The software enhancement fails due to lack of true commitment. Management sets no outlook from the development community regarding process improvement.

Process Improvement Learning Curve



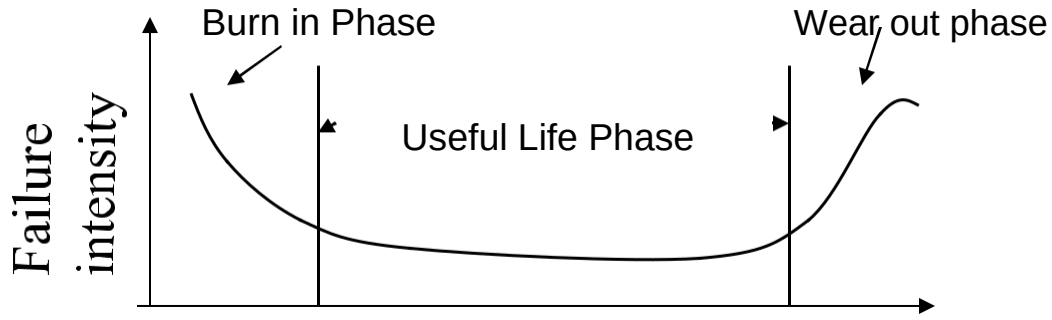
A **Learning Curve** is a graph depicting relationship between learning and amount of effort. It's very difficult to climb steep curve or mountain and takes lots of efforts and time.

"A Steep Learning Curve" refers to something difficult to learn.

Software Characteristics

- **Software is developed**
  It is not manufactured. It is not something that will automatically roll out of an assembly line. It ultimately depend on individual skill and creative ability

- **Software does not Wear Out**
  Software is not susceptible to the environmental melodies and it does not suffer from any effects with time.



- **Software is Highly Malleable**
  In case of software one can modify the product itself rather easily without necessary changes.

Software Myths(Management Perspectives)

Management may be confident about good standards and clear procedures of the company.

Software Myths(Management Perspectives)

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

Software Myths(Management Perspectives)

Addition of more software specialists, those with higher skills and longer experience may bring the schedule back on the track!

Software Myths (Management Perspectives)

Software is easy to change

The reality is totally different.

Software Myths (Management Perspectives)

Computers provide greater reliability than the devices they replace

This is not always true.

Software Myths (Customer Perspectives)

A general statement of objectives is sufficient to get started with the development of software. Missing/vague requirements can easily be incorporated/detailed out as they get concretized.

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

Software Myths (Customer Perspectives)

Software with more features is better software

Software can work right the first time

Both are only myths!

Software Myths (Customer Perspectives)

Once the software is demonstrated, the job is done.

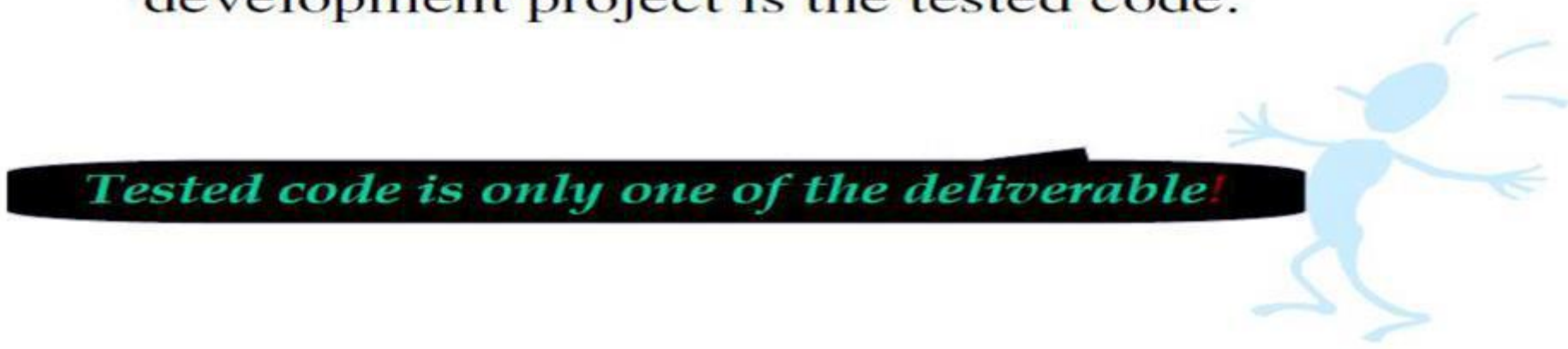
Usually, the problems just begin!

Software Myths (Developer Perspectives)

Software quality can not be assessed before testing.

Software Myths (Developer Perspectives)

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

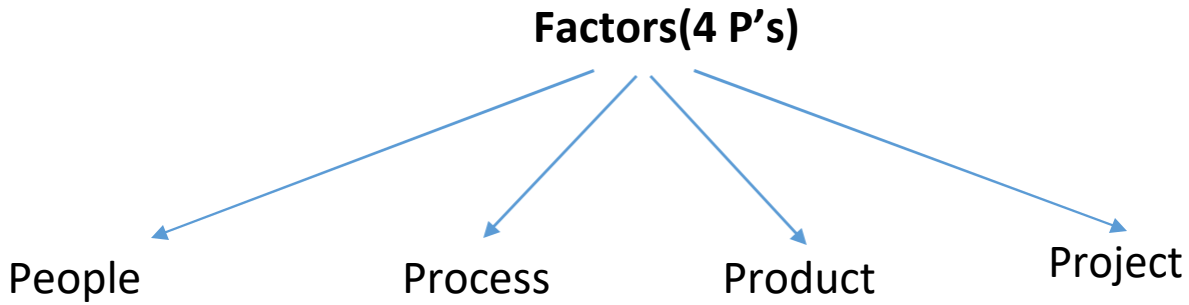


Software Myths (Developer Perspectives)

Aim is to develop working programs

Role of Management in Software Development

To ensure that **software** products and **software engineering** services are delivered efficiently, effectively, and to the benefit of stakeholders.



Role of Management in Software Development Based on Dependency