

Software Engineering

Muhammad Rabee Shaheen, Ph.D

Course Overview

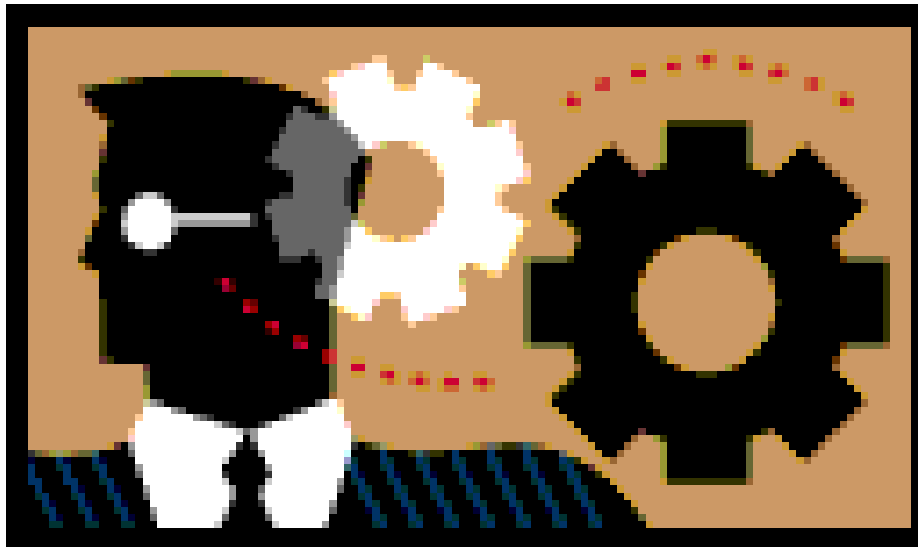
- **Introduction**
- **Software Life Cycle Models**
- **Software Requirements (Analysis & Specifications)**
- **Software Project Planning**
- **Software Design**
- **Software Metrics**
- **Software Reliability**
- **Software Testing**
- **Software Maintenance**

Today's Topics

- **Introduction**
- **Software Crisis**
- **Software Costs**
- **Software Engineering**
- **Software Engineers**
- **Software and Programs**
- **Documentation**
- **Operating Procedures**
- **Software Products**
- **Product Specifications**
- **FAQs about Software Engineering**
- **Software Myths**
- ...

Introduction

- **Software yesterday, today, tomorrow**
 - **What is the difference?**



Introduction

- **Single processor, alphanumeric output, input from linear source**
- **GUI**
- **Client-server architecture**
- **Several processors**
- **Different OS**
- **Geographically distributed machines**

Software complexity?

Introduction

- **Any challenge?**
 - **Abreast of new technology**
 - **Development backlogs**
 - **Cope with people issues**
 - **“Guru of project” concept**
 - **Do all employees like the change? (automation)**

Introduction

“31% of projects 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” IBM Report

Software Costs

- **Software costs often dominate computer system costs. The costs of software on a PC are often greater than the hardware cost.**
- **Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs.**

Software Crisis

- **Software:**
 - still come late
 - exceed budget
 - full of residual faults
- **Some failures:**
 - Y2K: million have been spent to handle this practically non-existent problem
 - Star wars (Patriot missile) è 28 U.S. soldiers, due to a small timing error in the system's clock

Software Crisis

- Some failures ...
 - In 1996, a US consumer group embarked on an 18-month, \$1million project to replace its customer database. *The new system was delivered on time but did not work as promised. As a result new tram was brought in to rebuild the system!*
 - Ariane-5 space rocket, \$7000 M, over a 10 years:
 - was destroyed after 39 seconds of its launch!!
 - Conversion error: 64-bit to 16-bit format

Software Crisis

- **Some 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”*
 - **Windows XP: released on October 25, 2001**
 - On the same day company posted 18 MB:
 - Compatibility patches (for bug fixes)
 - Compatibility updates
 - Enhancement
 - 2 patches fixed important security holes

Software Crisis

There are serious problems in the cost, timeliness, maintenance and quality of many software products

**So what should we do?
How could we react?**

Software Engineering

- SE has the objective of solving these problems by producing good quality, maintainable software, on time, within budget.

“The establishment and use of sound engineering principles in order to obtain economically developed software that is reliable and works efficiently on real machines” 1st SE conf. 1968

“A discipline whose aim is the production of quality software, software that is delivered on time, within budget and that satisfies its requirements” Schach

Software Engineers

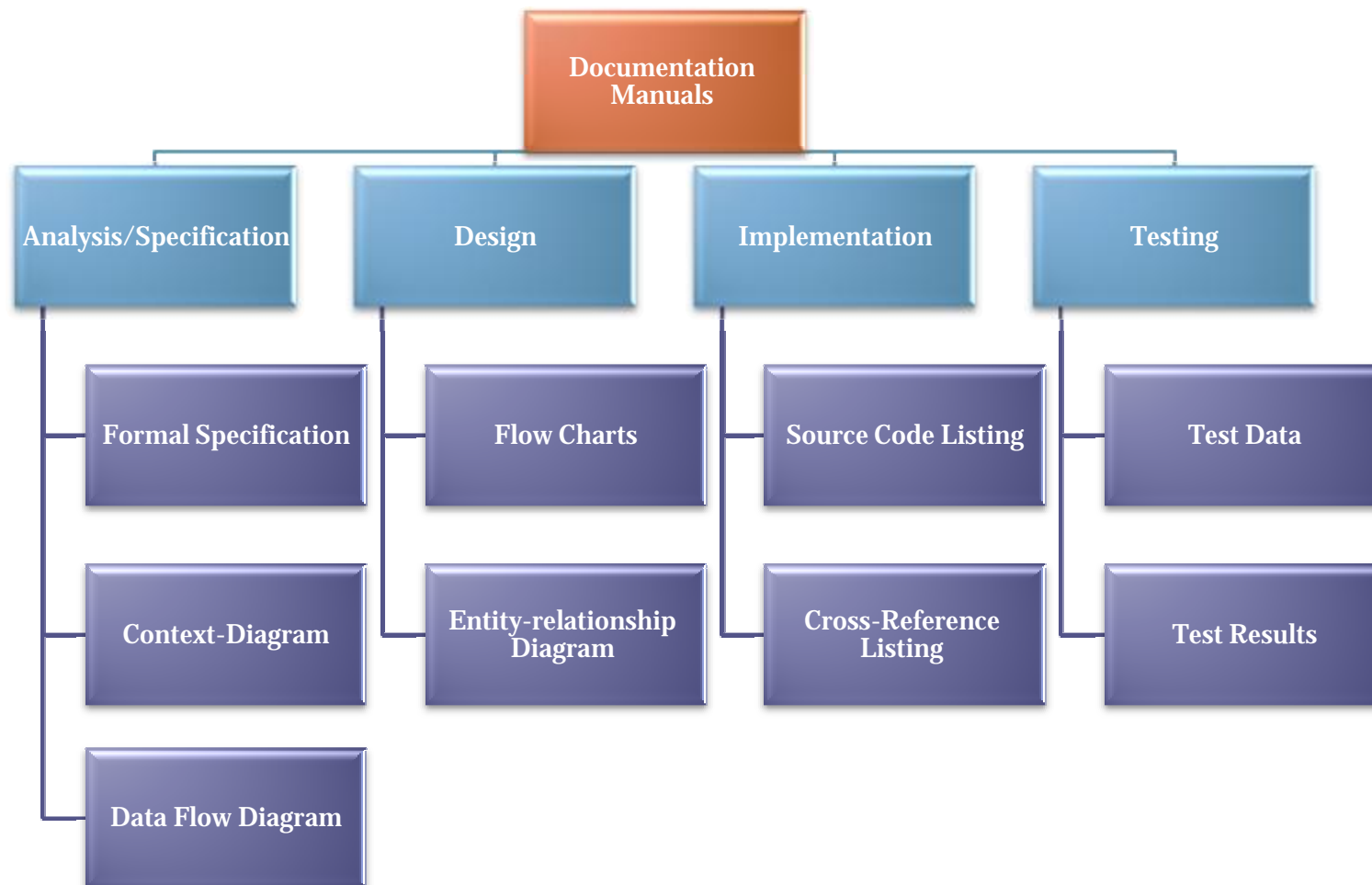
- Software ***engineers*** must:
 - Adopt a systematic and organized approach to their work
 - Use appropriate tools and techniques depending on the problem to be solved and the development constraints
 - Use the resource available

Software, Programs

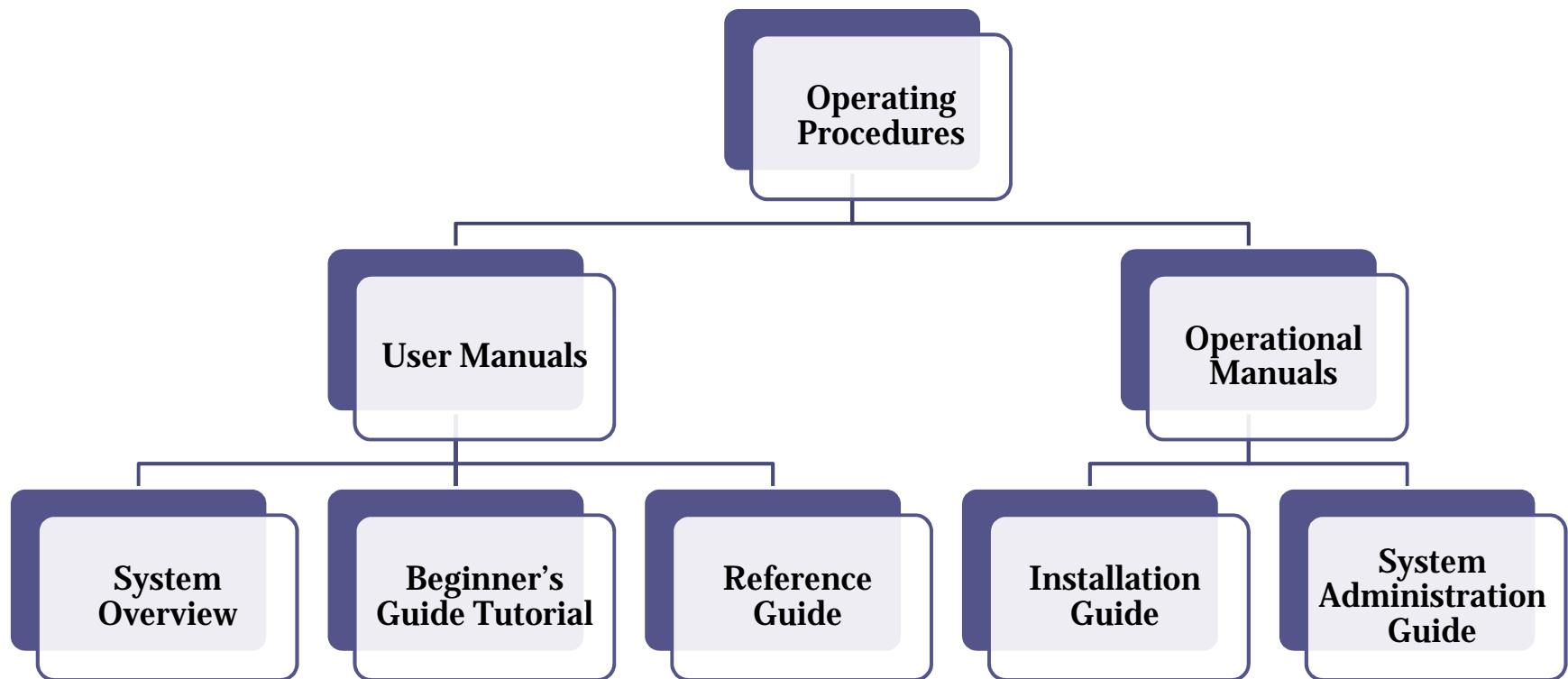


Software consists of programs, documentation of any facet of the program and the procedures used to setup and operate the software system. While program is source code

Documentation?



Operating Procedures



Software Products

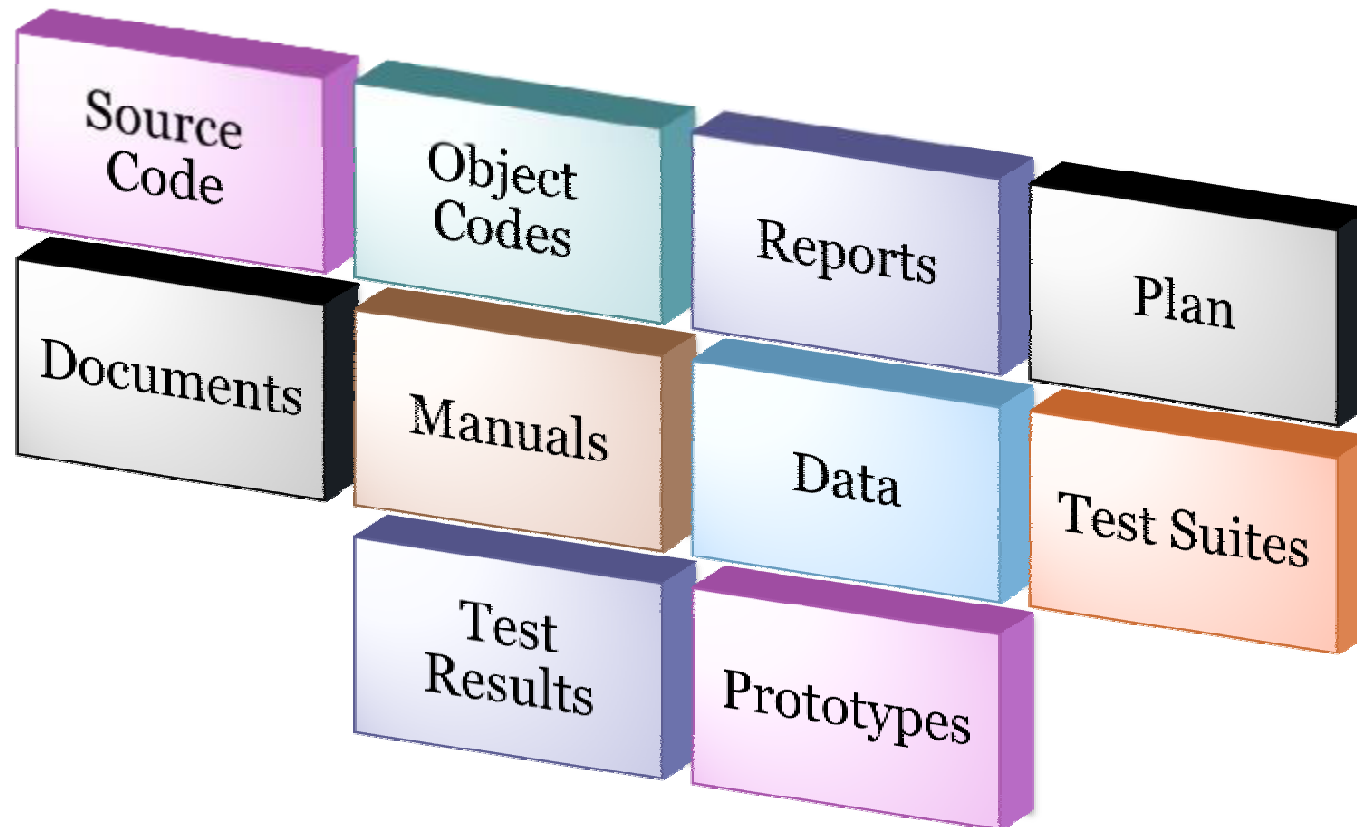
- **Generic ?**
- **Bespoke (Custom) ?**

Product Specifications

- **Generic:**
 - The specification of what the software should do is owned by the software developer and decisions on software change are made by the developer.
- **Customized:**
 - The specification of what the software should do is owned by the customer for the software and they make decisions on software changes that are required.

Software Product

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



Software Myths

- 1. Software easy to change**
- 2. Computers provide greater reliability than the devices they replace**
- 3. Testing software or “proving” software correct can remove all the errors**
- 4. Reusing software increases safety**
- 5. Software can work right the first time**
- 6. Software can designed thoroughly enough to avoid most integration problems**

Software Myths

- 7. Software with more features is better software**
- 8. Addition of more software engineers will make up the delay**
- 9. Aim is to develop working programs**

Software Process

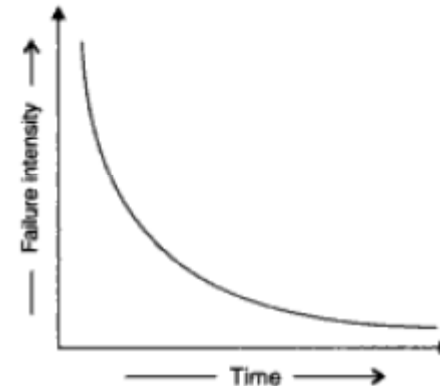
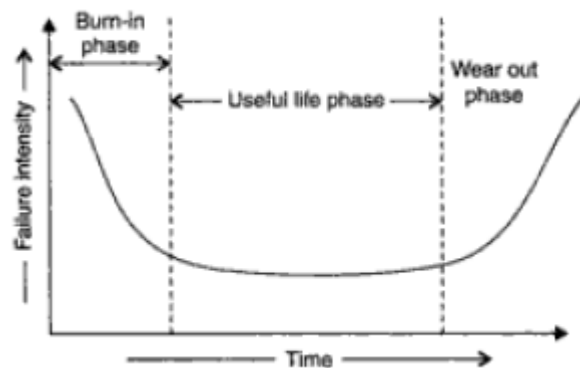
- **SP: the way in which we produce software**
- **SP: help the developers to use the best technical and managerial practices to successfully complete their projects**
- **SP is a way to improve the quality, productivity, predictability of the software development and maintenance efforts**

Software Process Activities

- **Software specification**, where customers and engineers define the software that is to be produced and the constraints on its operation.
- **Software development**, where the software is designed and programmed.
- **Software validation**, where the software is checked to ensure that it is what the customer requires.
- **Software evolution**, where the software is modified to reflect changing customer and market requirements.

Software Characteristics

- **Software does not wear out (hardware vs. software)**



- **Software is not manufactured (just copies!)**
- **Reusability of components**
- **Software is flexible**

Software Applications

- **System Software:** compilers, OS, drivers
- **Real-time Software:** used to monitor, control and analyze real world events (weather,..)
- **Embedded Software:** ROM, aircraft, automobile,..
- **Business Software:** employee management, account management, ...
- **Personal Computer Software:** MS-office, graphics,...

Software Applications

- **Artificial Intelligence Software:** expert systems, neural network, signal processing,...
- **Web Based Software:** CGI, HTML, Perl,...
- **Engineering and Scientific Software:** CAD, SPSS, MATLAB, Circuit analyzers, ...

Good Software

| Product characteristic | Description |
|----------------------------|--|
| Maintainability | Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment. |
| Dependability and security | Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system. |
| Efficiency | Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc. |
| Acceptability | Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use. |

FAQs about SE

| Question | Answer |
|---|---|
| What is software? | Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market. |
| What are the attributes of good software? | Good software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable. |
| What is software engineering? | Software engineering is an engineering discipline that is concerned with all aspects of software production. |
| What are the fundamental software engineering activities? | Software specification, software development, software validation and software evolution. |
| What is the difference between software engineering and computer science? | Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software. |
| What is the difference between software engineering and system engineering? | System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering. Software engineering is part of this more general process. |

FAQs about SE

| Question | Answer |
|--|--|
| What are the key challenges facing software engineering? | Coping with increasing diversity, demands for reduced delivery times and developing trustworthy software. |
| What are the costs of software engineering? | Roughly 60% of software costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs. |
| What are the best software engineering techniques and methods? | While all software projects have to be professionally managed and developed, different techniques are appropriate for different types of system. For example, games should always be developed using a series of prototypes whereas safety critical control systems require a complete and analyzable specification to be developed. You can't, therefore, say that one method is better than another. |
| What differences has the web made to software engineering? | The web has led to the availability of software services and the possibility of developing highly distributed service-based systems. Web-based systems development has led to important advances in programming languages and software reuse. |

Frequent Terminologies in SE

- **Deliverables**

- Generated during software development
- E.g., source code, user manuals, operating procedures,...

- **Milestones**

- Events that are used to ascertain the status of the project
- E.g., Finalization of specification is a milestone, Completion of design documentation, ...

Frequent Terminologies in SE ...

- **Product**

- is what is delivered to the customer
- includes source code, specification document, manuals, documentation,...
- Basically, a set of deliverables only

- **Process**

- way in which we produce software
- collection of activities that leads to (a part of) a product.

Frequent Terminologies in SE ...

- **Measures** (Measurement, Metrics)
 - **Measure:** a quantitative indication of the extent, dimension, size, capacity, or reliability of some attributes of a product or process. *E.g., #errors in the review of a single module*
 - **Measurement:** the act of evaluating a measure. *E.g., #modules are investigated to collect measures of the #errors in each module*
 - **Metrics:** relating the individual measures in some way. *E.g., avg. #errors found per module*

Frequent Terminologies in SE ...

- **Software Process & Product Metrics**
 - **Software metrics:** used to quantitatively characterize different aspects of software process or software products.
 - **Process metrics:** quantify the attributes of software development process and environment. *E.g., productivity, quality, failure rate, ...*
 - **Product metrics:** are measures for the software product. *E.g., size, reliability, complexity, functionality,...*

Frequent Terminologies in SE ...

- **Productivity**

- defined as the rate of output, production per unit of effort
- Output achieved with regard to the time taken, but irrespective of the cost incurred.
- unit of measure:
 - Quantity of output: e.g., LOC produced
 - Time: measured in days or months
- Unit of effort: most appropriate unit of effort is Person Months (PMs), *i.e., #persons involved for specified months*

Frequent Terminologies in SE ...

- **Productivity...**
 - Productivity may be measured as LOC/PM
- **Module**
 - a Fortran subroutine
 - an Ada Package
 - “procedures & functions” in Pascal & C
 - “C++, Java class, Java packages”
 - a work assignment for an individual developer

Frequent Terminologies in SE ...

- **Component**
 - an independently deliverable piece of functionality providing access to its services through interfaces
- **Generic & Customized Software Products**
 - *Previously discussed*

Role of Management in Software Development (SD)

- **Four factors:**

- **People**
- **Product**
- **Process**
- **Project**

Role of Management in Software Development (SD)

- **Four factors:**

- **People**
- **Product**
- **Process**
- **Project**

SD requires good managers

Good managers understand people psychology

Provide good leadership

Cannot ensure the success of project, but can increase the probability of success

Priority: selection, training, compensation, career development, work culture, ...

Role of Management in Software Development (SD) ...

- **Four factors:**

- **People**
- **Product**
- **Process**
- **Project**

What do we want to deliver to the customer?

Define objectives & scope of work (requirements)

Discussion of alternative solutions

Select best approach within constraints imposed by delivery deadline, budget, personnel availability,...

Define estimated cost, Development time, schedule,...

Role of Management in Software Development (SD) ...

- **Four factors:**

- **People**
- **Product**
- **Process**
- **Project**

Way in which we produce software

Provides framework from which a comprehensive plan for software development can be established

Several life cycle models & process improvement models

CMM (Capability Maturity Model) , a standard for process framework

Role of Management in Software Development (SD) ...

- **Four factors:**

- **People**
- **Product**
- **Process**
- **Project**

A planning is required to monitor the status of SD

A planning is required to control the complexity

In a successful project, we must understand what can go wrong & how to do it right

Define concrete requirements & freeze them

Changes should not be incorporated to avoid software surprises, because they are always risky!

Covered Topics

- **Software Crisis**
- **Software Costs**
- **Software Engineering**
- **Documentation**
- **Software Products**
- **Product Specification**
- **Software Myths**
- **Software Process Activities**
- **Software Application Types**
- **Good Software**
- **FAQs**
- **Software Engineering Terminologies**
- **Management in SD**