UNIT I

THE SOFTWARE PROBLEM & SOFTWARE PROCESS

1. SOFTWARE PROBLEM

1.1 COST,SCHEDULE AND QUALITY

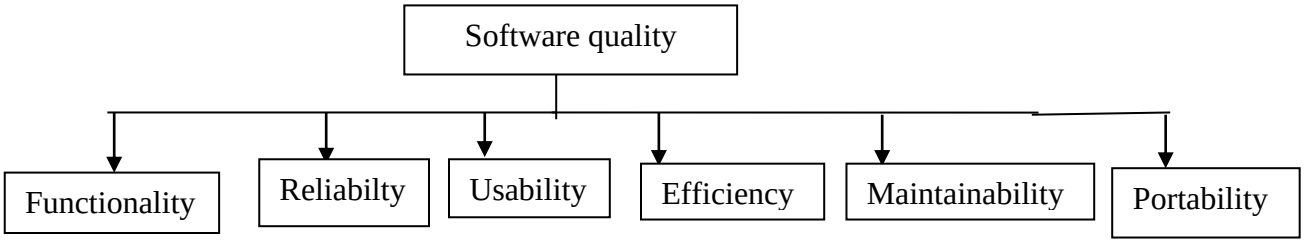
COST

- In the industrial-strength software domain, there are three basic forces -cost, schedule and quality. These three parameters drive and define a software project.
- Industrial-strength software is very expensive.
- The productivity in the software industry for writing fresh code generally ranges from few hundred to about 1000+ LOC (Lines Of Code) per person-month.
- Software companies often charge the client for whom they are developing the software between $3000-$15000 per person-month.
- With a productivity of 1000 LOC per person-month, it means that each line of delivered code costs between $3 and $15.Even small projects can easily end up with software of 50,000 LOC.With this productivity, such a software project will cost between $150,000 and $750,000.

SCHEDULE

- Schedule is an important factor in many projects.
- Business trends are dictating that the time-to-market of a product should be reduced, that is, the cycle time from concept to delivery should be small.
- Software needs to be developed faster and within the specified time.

QUALITY



Software quality attributes

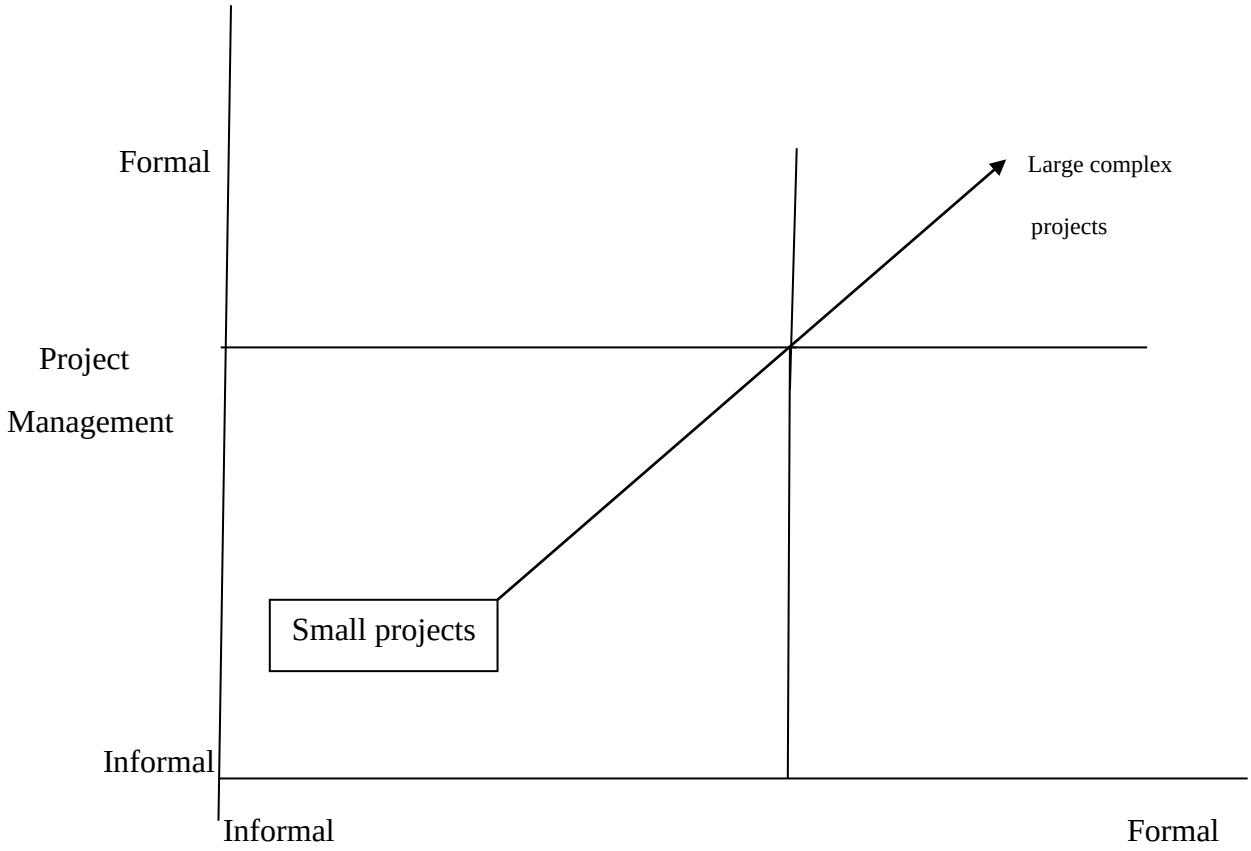
The international standard on software product quality suggests that software quality comprises six main attributes:

- **Functionality:** The capability to provide functions which meet stated and implied needs when the software is used.
- **Reliabilty:** The capability to provide failure-free service.
- **Usability:** The capability to be understood, learned,and used.
- **Efficiency:** The capability to provide appropriate performance relative to the amount of resources used.
- **Maintainability:** The capability to be modified for purposes of making corrections,improvements,or adaptation.
- **Portability:** The capability to be adapted for different specified environments without applying actions or means other than those provided for this purpose in the product.

1.2 SCALE AND CHANGE

SCALE

- Most industrial-strength software systems tend to be large and complex,requiring tens of thousands of LOC.
- Development of a large system requires a different set of methods compared to developing a small system, as the methods that are used for developing small systems do not scale up to large systems.
- Methods used to develop programs for small scale software is different from methods for developing programs for large software.
- In small projects, informal methods for development and management can be used. For large projects, both have to be much more rigorous



The problem of scale

CHANGE

- Software has to be changed even after it has been deployed.
- Business is changing very rapidly in modern age. As business changes, the software supporting it needs to be changed.
- Rapid change in software has a special impact on it. There are lot of expectations from changed software. So, it is one of the challenges in Software Engineering.
- By applying SE principles, software of high quality and productivity can be produced but these principles are not useful for accommodating even for small changes.

Difference between Student Software and Industrial-strength Software

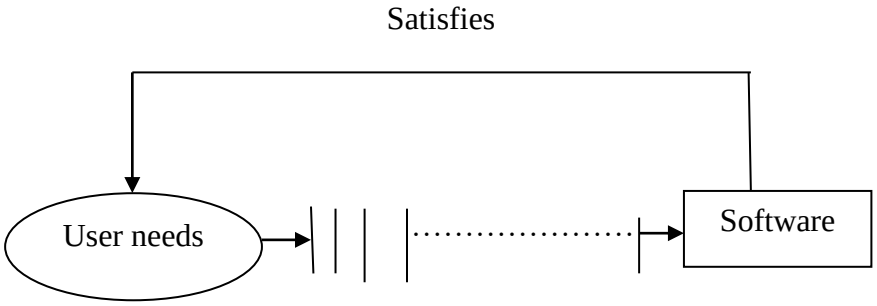
Student Software	Industrial-strength Software
It is a kind of software system developed by students.	It is a kind of software system developed in a company by a team of software professionals for their clients.
It can be developed in a random manner.	It is developed in phases like requirements analysis, design, coding, testing and maintenance.
There is no business involved.	Clients always pay for developed software systems. The business of company depends on working model of software system.
It is simply for demonstration purpose and generally not used for solving any real problem of any organization.	It is used to solve some problems of clients. Example: Managing finance or railway reservation system.
It is not designed by concerning quality factors that is, portability, usability and so on.	This software needs high quality such as reliability, user friendly and so on.
No significant activity depends on proper functioning. Bugs/Errors are acceptable.	Many important activities depend on correct functioning of this software system.

2. SOFTWARE PROCESS

2.1 PROCESS AND PROJECT

PROCESS

- A process is a sequence of steps performed for a given purpose.
- While developing (industrial strength) software,the purpose is to develop software to satisfy the needs of some users or clients.



Basic problem

- The role of process increases due to additional goals and many processes can achieve the basic goal of developing software to achieve high Q&P.It is this goal that makes designing a process a challenge.
- A process is a dynamic entity which captures the actions performed. Process specification is a description of the process which presumably can be followed in some project to achieve the goal for which the process is designed.
- The actual process is what is actually done in the project. A process model specifies a general process.
- A process is a means to reach the goals of high quality, low cost and low cycle time. A process model provides a process structure that is well suited for a class of projects.
- A process is specified at a high level as a sequence of stages. The sequence of steps for a stage is the process for that stage and is referred as a subprocess of the process.

PROJECT

- A project follows process. The desired end goal of delivering the software is achieved.

- In a project, a process specification may be used as the process the project plans to follow.

Difference between Software Process and Software Project

Software Process	Software Project
It is developed by individual user and used for personal use.	It is developed by multiple users and used by large number of people or customers.
It may be small in size and processing limited functionality. It defines framework for effective product generation.	It consists of multiple program codes, related documents such as SRS, designing documents, user manuals, test cases and so on.
Process is generally developed by process engineering.	It is developed by software engineers who are large in number and work in a team. Hence, systematic approach of developing software must be applied.
The output of process is product.	Product development occurs by following process
Example: Program developed for parsing the input.	Example: A word processing software.

Difference between Process and Project

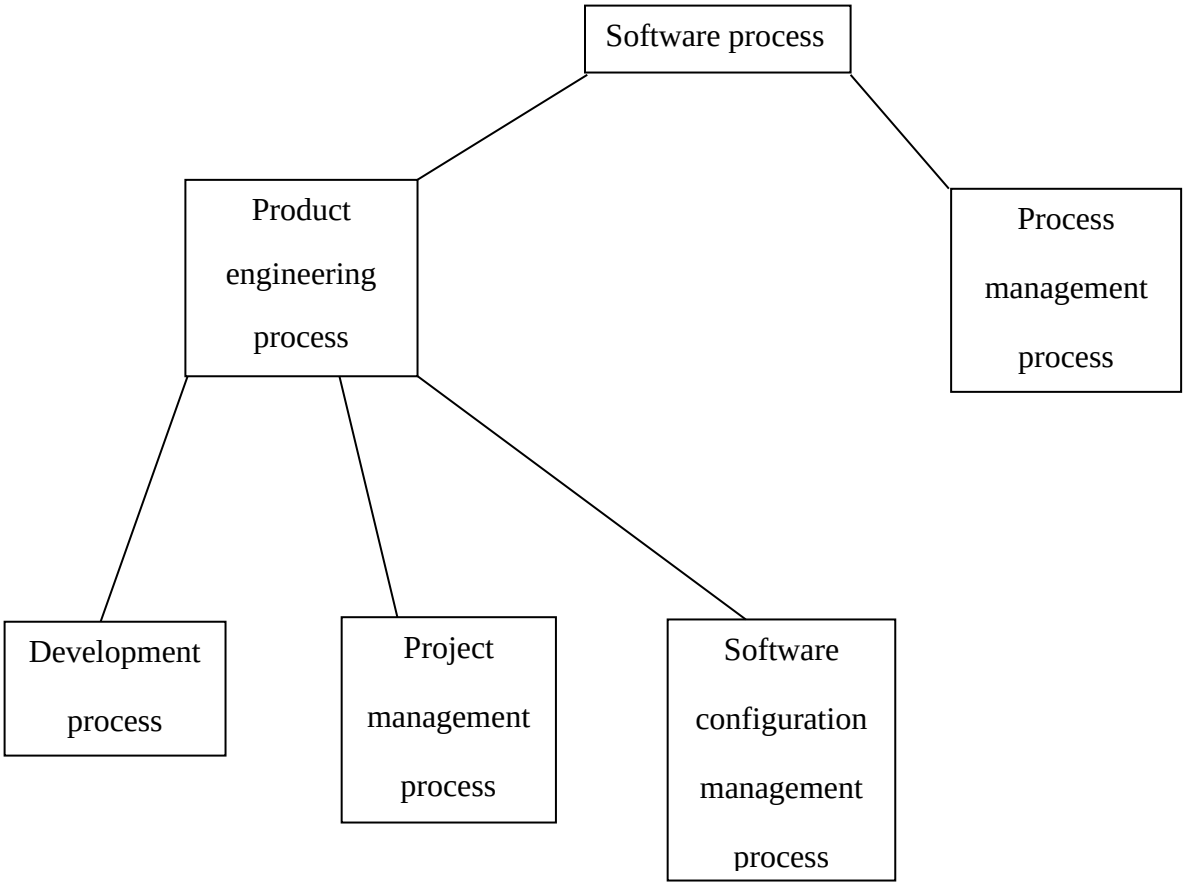
Process	Project
It has an objective that is typically defined	It has an objective or outcome to be

around the ongoing operation of the process. Example: Ongoing maintenance for motor vehicle.	accomplished and project ends when objective is accomplished. Example: Finding a replacement ignition switch that solves the problem with motor vehicles.
It is generally ongoing and does not normally have an end.	It has a beginning and an end.
It is a repetitive sequence of tasks and the tasks are known at the outset since it is repetitive.	The sequence of tasks is not normally repetitive and may not be known at the outset of project.

2.2 COMPONENT SOFTWARE PROCESSES

- The processes that deal with the technical and management issues of software development are collectively called the software process.
- There are two major components in a software process- a development process and a project management process.
- The development process specifies all the engineering activities that need to be performed. The management process specifies how to plan and control these activities so that cost, schedule, quality and other objectives are met.
- As development processes generally do not focus on evolution and changes, to handle them another process called software configuration control process is used. The objective is to primarily deal with managing change, so that the integrity of the products is not violated despite changes.
- These three constituent processes focus on the projects and the products and can be considered as product engineering process. The main objective is to produce the desired product.
- The basic objective of the process management process is to improve the software process. It means the capability of the process to produce quality goods at low cost is improved.
- The whole process of understanding the current process,analyzing its properties,determining how to improve and then affecting the improvement is done by process management process.

- In a typical project, development activities are performed by programmers, designers, testers, etc. The project management process activities are performed by the project management. Configuration control activities are performed by a group called configuration controller. The process management process activities are performed by software engineering process group(SEPG).

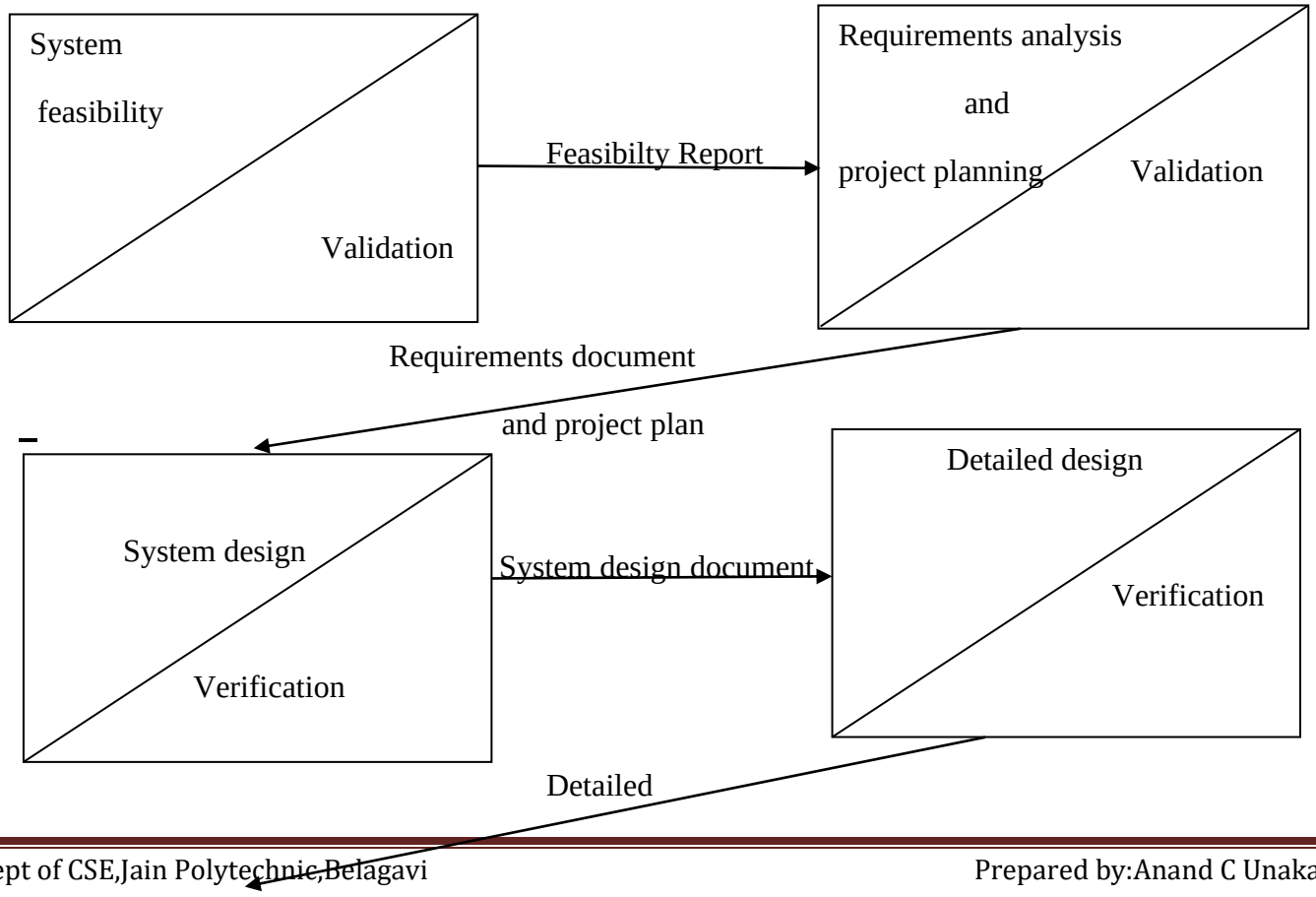


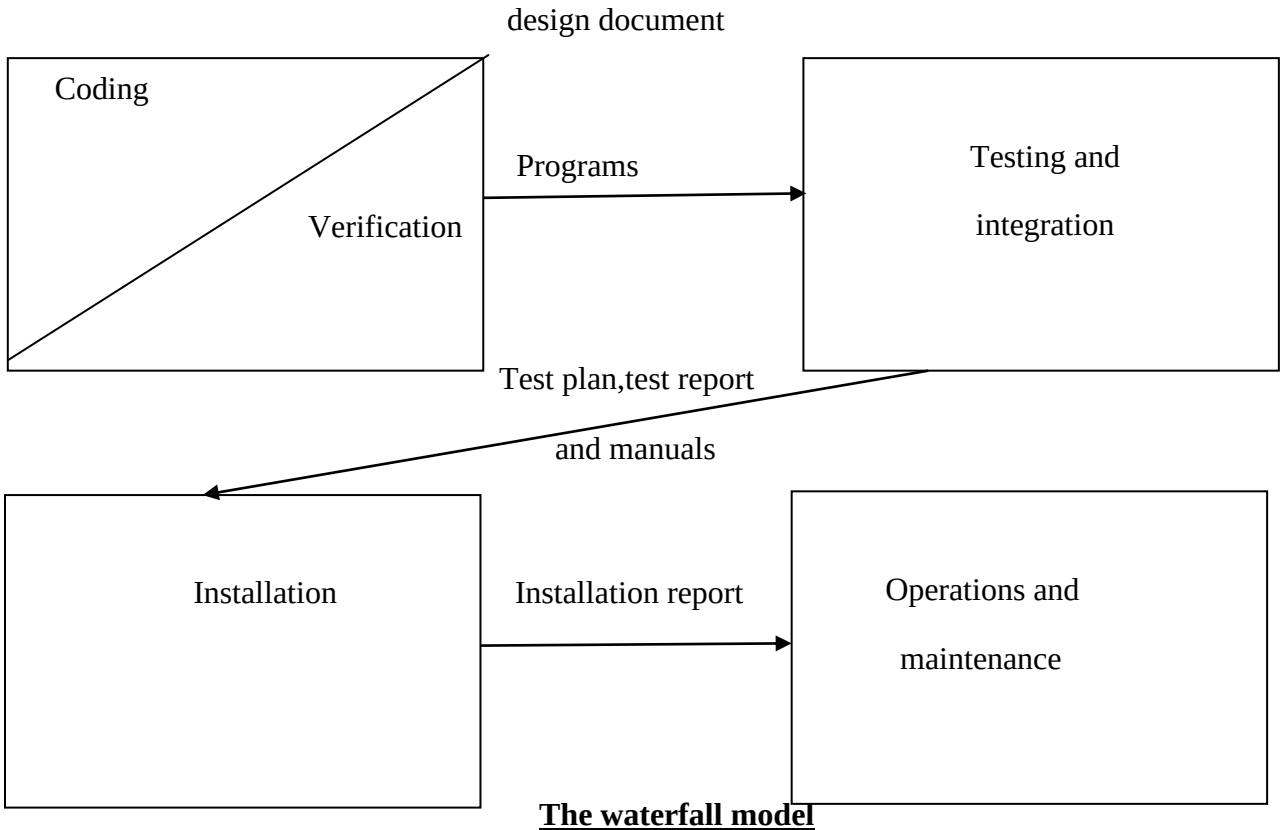
Software processes

2.3 SOFTWARE DEVELOPMENT PROCESS MODELS

- The goal is to produce a high-quality software product. It focuses on activities directly related to production of the software. Example: Design, Coding and Testing.
- A project's development process defines the tasks the project should perform and the order in which they should be done. The process drives a project and heavily influences the outcome.
- A process model specifies a general process as a set of stages in which a project should be divided,the order in which the stages should be executed and any other constraints and conditions on the execution of stages.
- Due to the importance of the development process,various models have been proposed.

WATERFALL MODEL





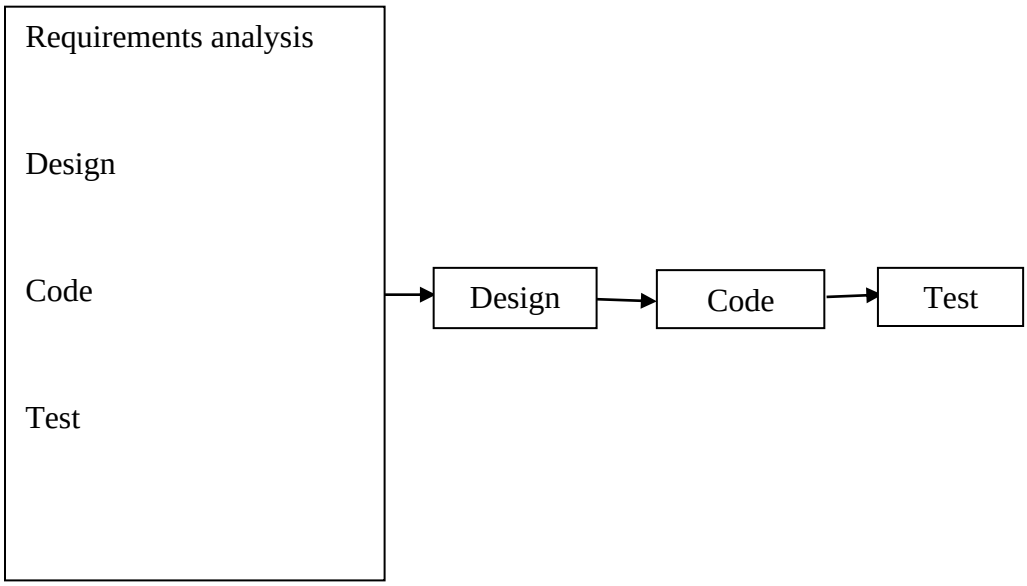
The waterfall model

- It is the simplest process model which states that the phases are organized in a linear order.
- In this model, a project begins with feasibility analysis. Upon successfully demonstrating the feasibility of a project, the requirements analysis and project planning begins.
- The design starts after the requirements analysis is complete.
- Coding begins after the design is complete. Once the programming is completed, the code is integrated and testing is done.
- Upon successful completion of testing, the system is installed. After this, the regular operation and maintenance of the system takes place.
- The basic idea behind the phases is separation of concerns.Each phase deals with a distinct and separate set of concerns.By doing this,the large and complex task of building the software is broken into smaller tasks of specifying requirements,doing design,etc.
- The requirements analysis phase is “analysis and planning.” Planning is a critical activity in software development. A good plan is based on the requirements of the system and should be made before later phases begin.

- Linear ordering of activities has some important consequences:
  - ✓ To clearly identify the end of a phase and the beginning of the next. Some certification mechanism has to be employed at the end of each phase. This is usually done by some verification and validation means that will ensure that the output of a phase is consistent with its input and that the output of the phase is consistent with the overall requirements of the system.
  - ✓ Need for certification is that each phase must have some defined output that can be evaluated and certified.When the activities of a phase are completed, there should be some product that is produced by that phase.The outputs of the earlier phases are called work products. For coding phase,the output is the code.
- The documents that should be produced in each project:
  - ✓ Requirements document
  - ✓ Project plan
  - ✓ Design documents(architecture,system,detailed)
  - ✓ Test plan and test reports
  - ✓ Final code
  - ✓ Software manuals (e.g.,user,installation,etc.)
- Advantages
  - ✓ Simple, easy to understand and use
  - ✓ Phases are processed and completed one at a time
  - ✓ Clearly defined stages
- Disadvantages
  - ✓ It is difficult to measure progress within stages
  - ✓ Cannot accommodate changing requirements
  - ✓ Poor model for long and ongoing projects

PROTOTYPING

- The goal of prototyping-based development process is to build a throwaway prototype to help understand the requirements.
- This prototype is developed based on the currently known requirements. Development of prototype undergoes design, coding and testing.
- By using this prototype, the client can get an actual feel of the system which enables clients to better understand requirements of desired system.
- Prototyping is an attractive idea for complicated and large systems in which there is no manual process or existing system to determine requirements.



Requirements analysis

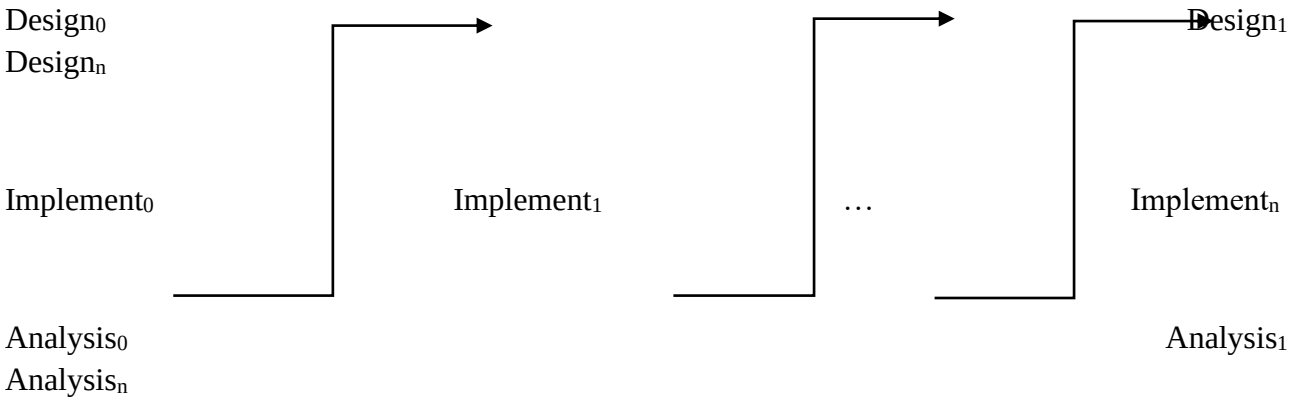
The prototyping model

- The development of the prototype starts when the preliminary version of the requirements specification document has been developed.
- At this stage, there is a reasonable understanding of the system and its needs and which needs are unclear or likely to change.
- After the prototype has been developed, the end users and clients are given an opportunity to use and explore the prototype.
- Based on their experience, they provide feedback to the developers regarding the prototype.
- Based on the feedback, the prototype is modified to incorporate some of the suggested changes that can be done easily and then the users and the clients are again allowed to use the system.Also,initial requirements are modified to produce final requirements specification which is used to develop software.
- This cycle repeats until the system is changed according to users' feedback.
- This model is suitable for projects where requirements are hard to determine and stated requirements are not trusted.

ITERATIVE DEVELOPMENT

- The software is developed in increments. Each increment adds some functional capability to the system until the full system is implemented.
- An example of this approach is iterative enhancement model.
- In the first step, a simple initial implementation is done for a subset of the overall problem. This subset contains some of the key aspects of the problem that are easy to understand and implement which forms a useful and usable system.
- A project control list is created that contains all the tasks in order that must be performed to obtain the final implementation. This list gives an idea of how far along the project is at any given step from the final system.

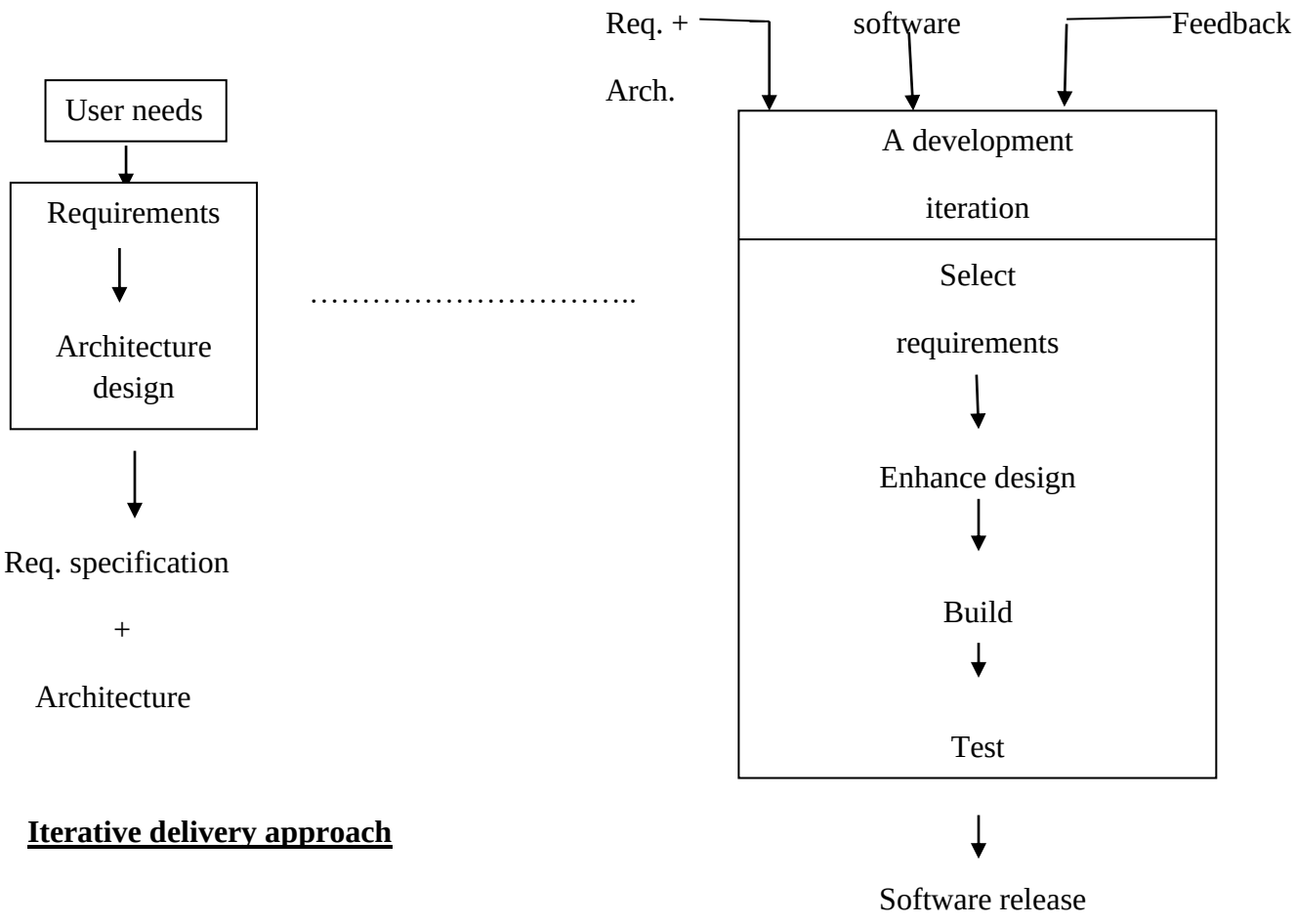
- Each step consists of removing the next task from the list, designing the implementation for the selected task, coding and testing the implementation, performing an analysis of the partial system obtained after this step, and updating the list as a result of the analysis. These three phases are called the design phase, implementation phase and analysis phase.
- This process is iterated until the project control list is empty when the final implementation of the system will be available. The project control list guides the iteration steps and keeps track of all tasks that must be done.



The iterative enhancement model

- Another common approach for iterative development is to do the requirements and the architecture design in a standard waterfall or prototyping approach but deliver software iteratively.
- The most time- and effort-consuming task is the building of the system which is done iteratively.
- At the start of each delivery iteration, requirements to be implemented in this release are decided and then the design is enhanced and code developed to implement the requirements.
- The iteration ends with delivery of a working software system providing some value to the end user.
- Selecting of requirements for an iteration is done primarily based on the value the requirement provides to the end users and how critical they are for supporting other requirements.

Existing



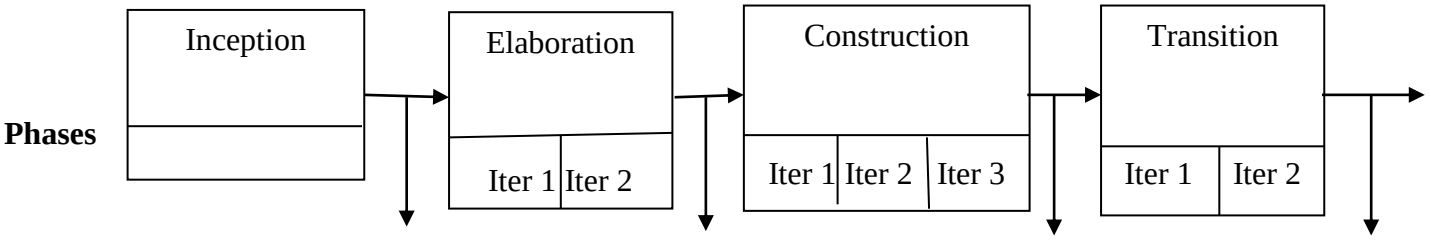
Iterative delivery approach

- Advantages
  - ✓ Progress of developed project can be measured
  - ✓ Less costly to change requirement
  - ✓ Testing and debugging is easier during smaller iterations
  - ✓ Results are obtained early and periodically
  - ✓ Better suited for large and mission-critical projects
- Disadvantages
  - ✓ More resources may be required
  - ✓ Not suitable for smaller projects
  - ✓ Defining increments may require definition of complete system

- ✓ Project progress is highly dependent upon risk analysis phase

RATIONAL UNIFIED PROCESS

- It is an iterative process model that was designed by Rational, now part of IBM. It was designed for object-oriented development using the Unified Modeling language(UML).
- In RUP, development of software is divided into cycles. Each cycle delivers a fully working system.
- Each cycle is executed as a separate project whose goal is to deliver some additional capability to an existing system. So, for a project the process for a cycle forms overall process.
- Each cycle is divided into four consecutive phases:
  - ✓ Inception phase
  - ✓ Elaboration phase
  - ✓ Construction phase
  - ✓ Transition phase



The RUP model

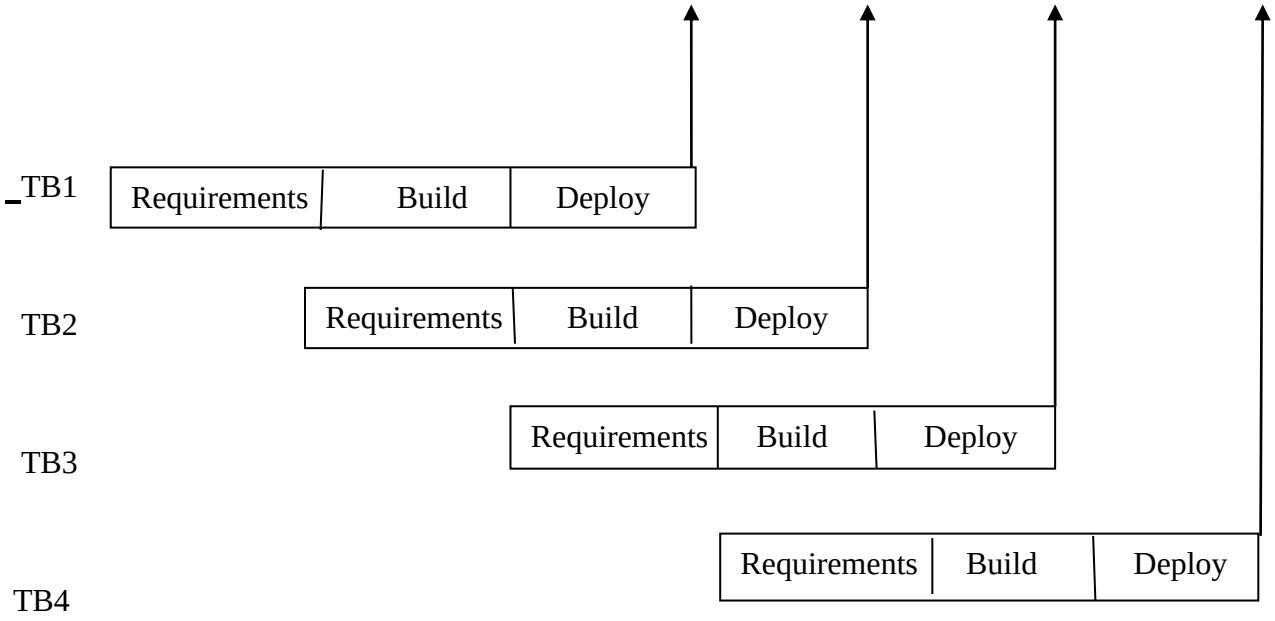
- Each phase has a distinct purpose and completion of each phase is a well-defined milestone in the project with some clearly defined outputs.
- The purpose of inception phase is to establish the goals and scope of the project and completion of this phase is the lifecycle objectives milestone. This milestone should specify the vision and high-level capability of the eventual system. Base on the output of this phase, go/no-go decision may be taken.
- In the elaboration phase, the architecture of the system is designed, based on the detailed requirements analysis. The completion of this phase is the lifecycle architecture milestone. At the end of this phase, it is expected that most of the requirements have been identified and specified

and the architecture of the system has been designed so it addresses the technical risks identified in the earlier phase.

- In the construction phase, the software is built and tested. This phase results in the software product to be delivered, along with associated user and other manuals. It results in initial operational capability.
- In transition phase, software is moved from the development environment to the client's environment where it is to be hosted. The successful execution of this phase results in product release.
- In this model, each phase have multiple iterations in which an internal or external customer delivers some well-defined output.
- Advantages
  - ✓ This methodology emphasizes on accurate documentation.
  - ✓ It is proactively able to resolve the project risk that is associated with the client.
  - ✓ Very less need for integration as the process of integration goes on throughout the development process.
- Disadvantages
  - ✓ The software developer needs to be expert in their work to develop software under this methodology.
  - ✓ The development process in this methodology is very complex and not exactly organized.
  - ✓ Integration throughout the process of software development adds on the confusion during stages of testing.

TIMEBOXING MODEL

- In this model, the basic unit of development is a time box, which is of fixed duration. So, requirements or features to be built in a time box
- Each timebox is divided into sequence of stages. Each stage performs some clearly defined task for the iteration and produces a clearly defined output.
- Duration of each stage is approximately same. The model requires that there be a dedicated team for each stage.

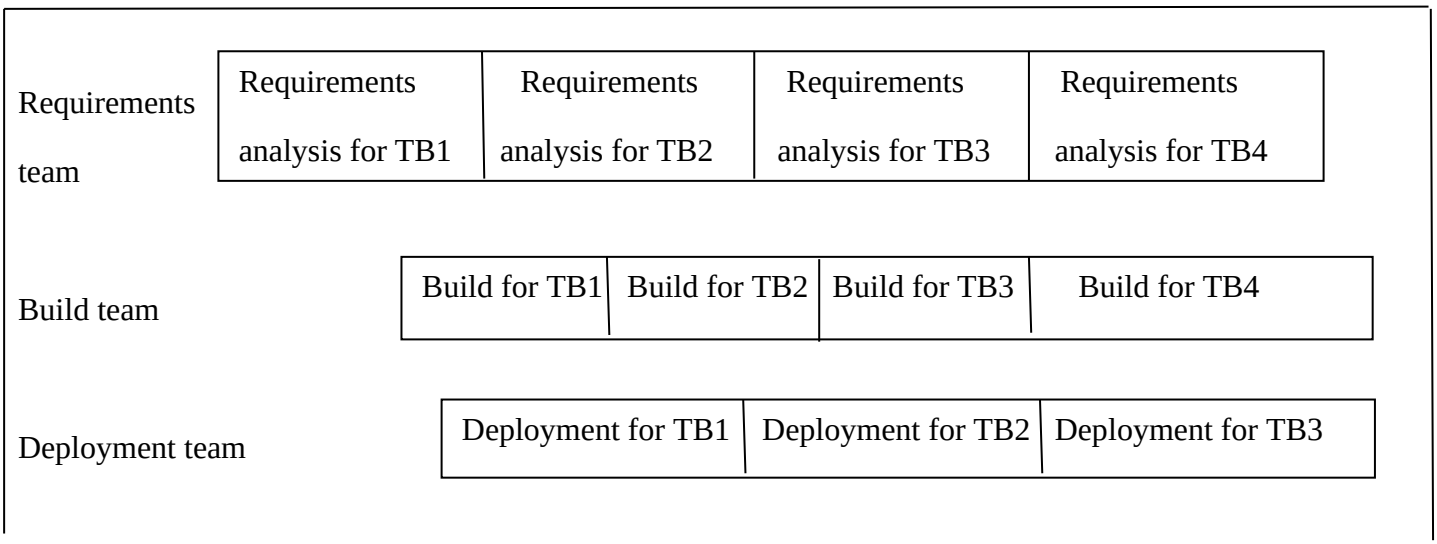


Executing the timeboxing process model

- Consider a time box consisting of three stages: requirement specification, build and deployment.
- The requirement stage is executed by its team of analysts and ends with a prioritized list of requirements to be built in this iteration along with a high-level design.
- The build team develops the code for implementing the requirements and performs the testing.
- The tested code is then handed over to the deployment team which performs predeployment tests and then installs the system for production use.
- These three stages can be done in approximately equal time in an iteration.
- With a time box of three stages, the project proceeds as follows. When the requirements team has finished requirements for timebox-1, the requirements are given to the build team for building the software. The requirements team then goes on and starts preparing the requirements for timebox-2 (TB2). When the build for timebox-1 (TB1) is completed, the code is handed over to the deployment team and the build team moves on to build code for requirements for TB2 and the requirements team moves on to doing requirements for timebox-3 (TB3).
- With a three stage time box, at most three iterations can be concurrently in progress. If the time box is of size T days, then the first software delivery will occur after T days. The subsequent deliveries will take place after every T/3 days. For example, if the time box duration T is 9 weeks, the first

delivery is made 9 weeks after the start of the project. The second delivery is made after 12 weeks, the third after 15 weeks and so on.

- Timeboxing is well suited for projects that require a large number of features to be developed in a short time around a stable architecture using stable technologies.



Tasks of different teams

- Advantages
  - ✓ Speeds up the development process and shortens delivery time
  - ✓ Well suited to develop project with a number of features in short time period
- Disadvantages
  - ✓ Project management becomes more complex
  - ✓ This model requires large team size to complete the project

EXTREME PROGRAMMING AND AGILE PROCESSES

- Agile Process
  - ✓ It consists of lightweight methods and these are people based methods.
  - ✓ It forces development team to focus on software itself.
  - ✓ It believes in iterative method.

- ✓ The aim of agile process is to deliver the working model of software to the customer at the earliest.

  Example:- Extreme Programming

- Principles of Agile Process

  - ✓ Customer Involvement

    The customer should be involved in the software development process. The customer specifies the software development and evaluates each iteration of the system.

  - ✓ Incremental Change

    The software is developed in increments. Each increment must be evaluated by the customer and also the customer can suggest any changes in the system and these changes must be implemented in increments.

  - ✓ Assume Simplicity

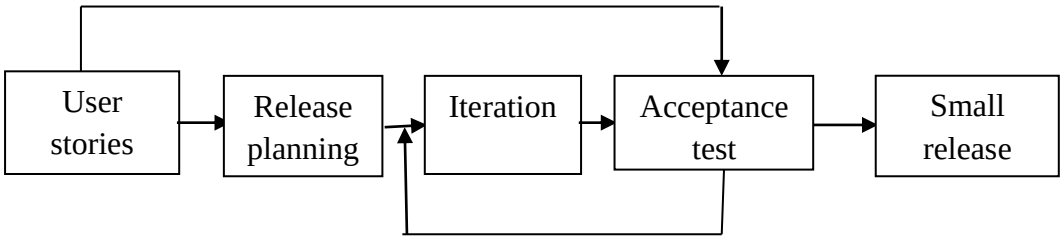
    The software system to be developed should be simple and the complexity of the system should be eliminated.

  - ✓ People are preferred than Process

    The skill of the people should be exploited. Instead of focusing on the process, the people should get liberty to develop their own working style.

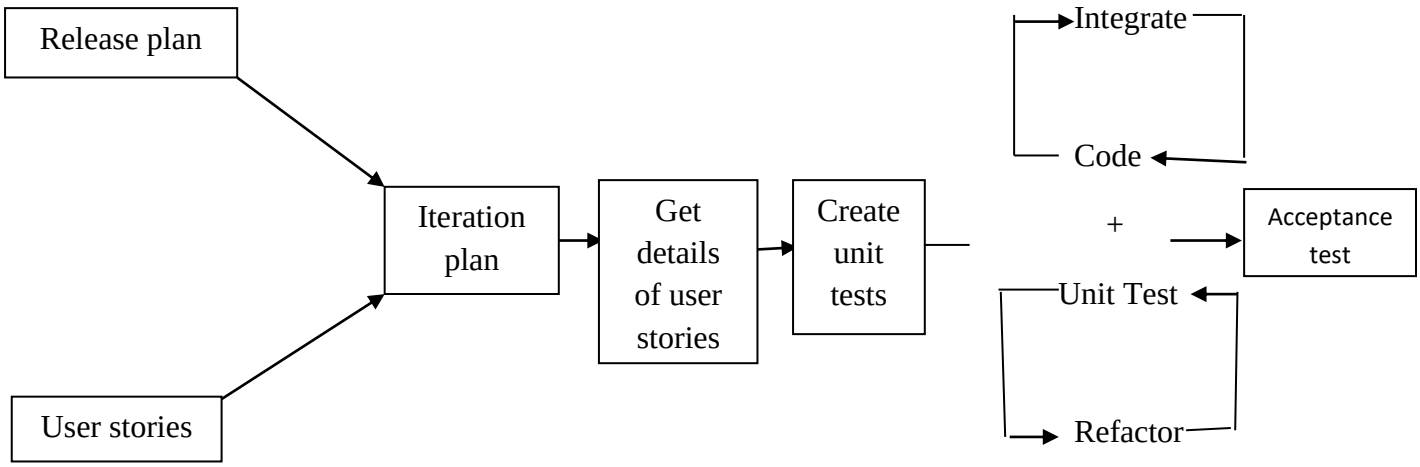
  - ✓ Embrace Change

    As the requirement changes over time, if any changes occur in the requirement or in design then the system should accommodate those changes.



Overall process in XP

- An XP project starts with user stories which are short (a few sentences) descriptions of what scenarios the customers and users would like the system to support. Each story is written on a separate card so they can be flexibly grouped.
- The development team estimates how long it will take to implement a user story. The estimates are rough, generally stated in weeks. Using these estimates and the stories, release planning is done that defines which stories are to be built in which system release and the dates of these releases.
- Development is done in iterations. Each iteration lasts for a few weeks. An iteration starts with iteration planning in which the stories to be implemented in this iteration are selected. high-value and high-risk stories are taken as higher priority and implemented in early iterations.
- Acceptance tests are also built from the stories which are used to test the software before the release.
- Bugs found during acceptance testing for iteration can form work items for the next iteration.



An iteration in XP

- The development approach used iteration has some unique practices:
  - ✓ It envisages that development is done by pairs of programmers.
  - ✓ It suggests that for building a code unit
    - ❖ Automated unit tests be written first before the actual code is written

- ❖ Code should be written to pass the tests

✓ It encourages simple solutions and change. It is expected that the design of the solution devised earlier may at some point become unsuitable for further development

- ❖ To handle this, it suggests that refactoring be done to improve the design and then use the refactored code for further development

✓ It encourages frequent integration of different units

- ❖ To avoid too many changes in the base code happening together, only one pair at a time can release their change and integrate into the common code base

USING PROCESS MODELS IN A PROJECT

- While developing (industrial strength) software, the purpose is to develop software to satisfy the needs of some users or clients and the project to be done at low cost and cycle time and deliver high-quality software.

- Given the constraints of the project, we would like to employ the process model that is likely to maximize the chances of delivering the software and achieve the highest Q&P. Hence, selecting a suitable development process model for a project is a key decision of project manager. We will take some examples.

- Suppose a small team of developers are given task of building a small auction site for a local university. The university administration is willing to spend some time at the start to help develop the requirements but their availability will be limited later. The team has been given 4 months to finish the project and deadline cannot be extended. The auction site will have some essential features and also other features without which system can function easily. With these constraints, it is clear that a waterfall model is not suitable as "all-or-nothing" risk is unacceptable due to inflexible deadline. The iterative enhancement model is also not right as it requires requirements analysis for each iteration and the users and clients are not available later. RUP is a suitable model as it allows iterations in each phase.

- Next example is that where the customers are in a highly competitive environment where requirements depend on what the competition is doing, and delivering functionality regularly is highly desirable. To reduce cost, the customer wants to outsource as much project work as possible to another team in another country. For this project, waterfall model is not suitable as requirements are not even known at the start. Iterative enhancement also may not work as it may not be able to deliver rapidly. XP will be hard to apply as it requires that the entire team including the customer be collocated. For this project, the timeboxing model seems to be best.

- Consider another project where a university wants to automate the registration process. It already has a database of courses and pre-requisites and a database of student records. As the requirements are well understood, so the waterfall model seems to be the best model.

3. PROJECT MANAGEMENT PROCESS

- Project Management Process specifies all activities that need to be done by the project management to ensure that cost and quality objectives are met.

- Its basic task is to ensure that once a development process is chosen, it is implemented optimally.

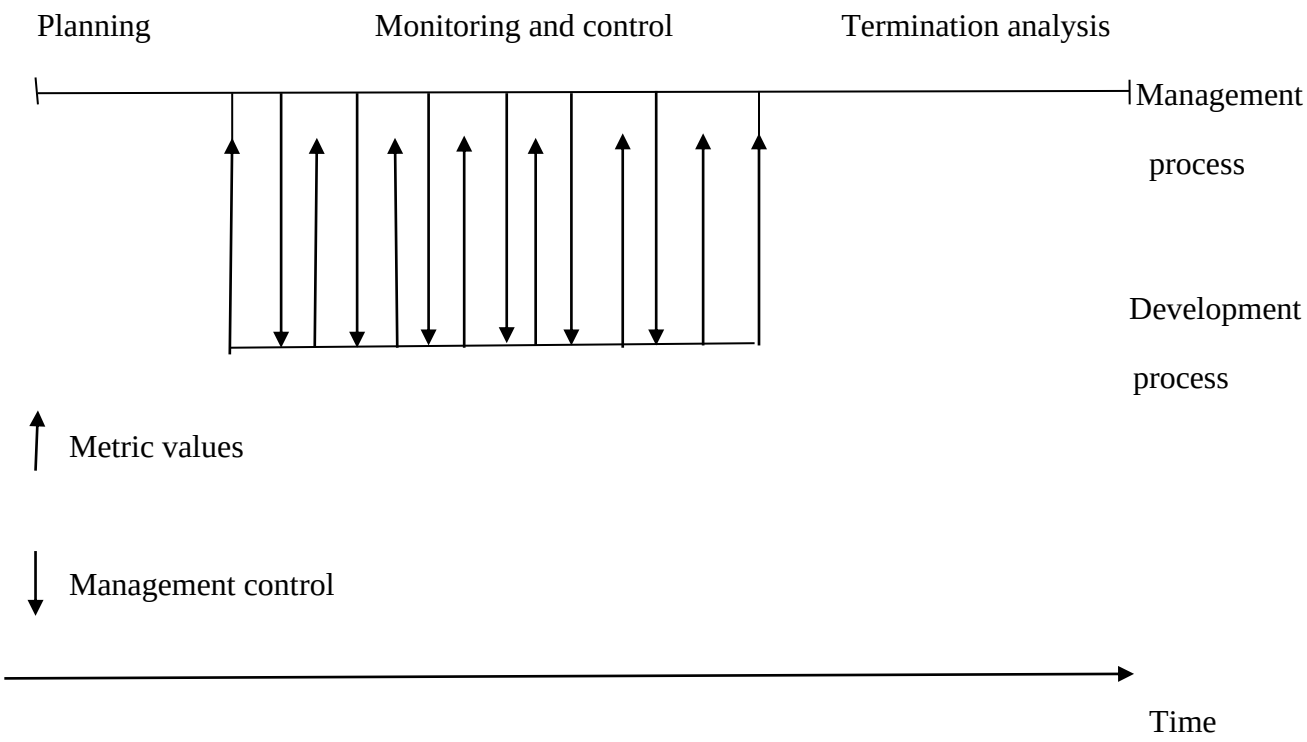
- The activities in the management process for a project is divided into three phases: planning, monitoring and control, termination analysis.

- Planning is the most critical project management activity. The goal of this phase is to develop a plan for software development due to which the objectives of the project can be met successfully and efficiently. A software plan is usually produced before the development activity begins and is updated as development proceeds and data about progress of the project become available. During planning, the major activities are cost estimation, schedule and milestone determination, project staffing, quality control plans and controlling and monitoring plans. Project planning forms the basis for monitoring and control.

- Project monitoring and control phase is the longest in terms of duration. It covers a major part of the development process. It includes all activities of the development to ensure that project objectives are met and the development proceeds according to the developed plan. Cost, schedule and quality are the major driving forces. Monitoring potential risks for the project is an important activity during the phase.

- Termination analysis is performed when the development process is over. It provides information about the development process and learn from the project in order to improve the process. This phase is also called “postmortem analysis”.

- The temporal relationship between the management process and the development process is shown in the figure.



Temporal relationship between development and management process

- It is an idealized relationship showing that planning is done before development begins and termination analysis is done after development is over.
- As shown in figure, during the development, from the various phases of the development process, quantitative information flows to the monitoring and control phase of the management process which uses the information to exert control on the development process.