Unit – II

[Software Process and SDLC Models]

1.Software Process: Process and project

Process :

A process is the sequence of steps executed to achieve a goal. A process is defined by cycles. Similar to a project, a process also has a beginning, middle, and end; however, this cycle repeats itself over an average period of time.

Project :

A project is defined by a fixed time, scope, and resources. When implementing a project, the goal is to execute change, usually drastic, and to incorporate that change into the day-to-day processes of the company.

Both projects and process are important for running and improving a business. However, depending on the end result trying to be achieved, one may be more suited than the other.

2.Component software process:

- The processes that deal with the technical and management issues of software development are collectively called the software process.
- As a software project will have to engineer a solution and properly manage the project, there are clearly 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, whereas the management process specifies how to plan and control these activities so that cost, schedule, quality, and other objectives are met. Effective

development and project management processes are the key to achieving the objectives of delivering the desired software satisfying the user needs, while ensuring high productivity and quality.

- As development processes generally do not focus on evolution and changes, to handle them another process called software configuration control process is often used. The objective of this component process is to primarily deal with managing change, so that the integrity of the products is not violated despite changes.

- product engineering processes, their main objective is to produce the desired product.

- The basic objective of the process management process is to improve the software process. By improvement, we mean that the capability of the process to produce quality goods at low cost is improved. For this, the current software process is studied, frequently by studying the projects that have been done using the process. The whole process of understanding the current process, analyzing its properties, determining how to improve, and then affecting the improvement is dealt with by the process management process.

- The relationship between these major component processes is shown in Figure 2.2. These component processes are distinct not only in the type of activities performed in them, but typically also in the people who perform the activities specified by the 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 process activities are performed by a group generally called the configuration controller; and the process management process activities are performed by the software engineering process group (SEPG).

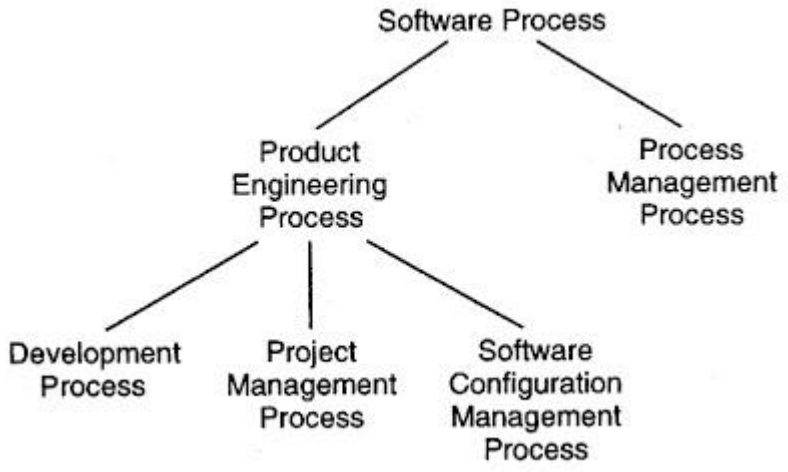


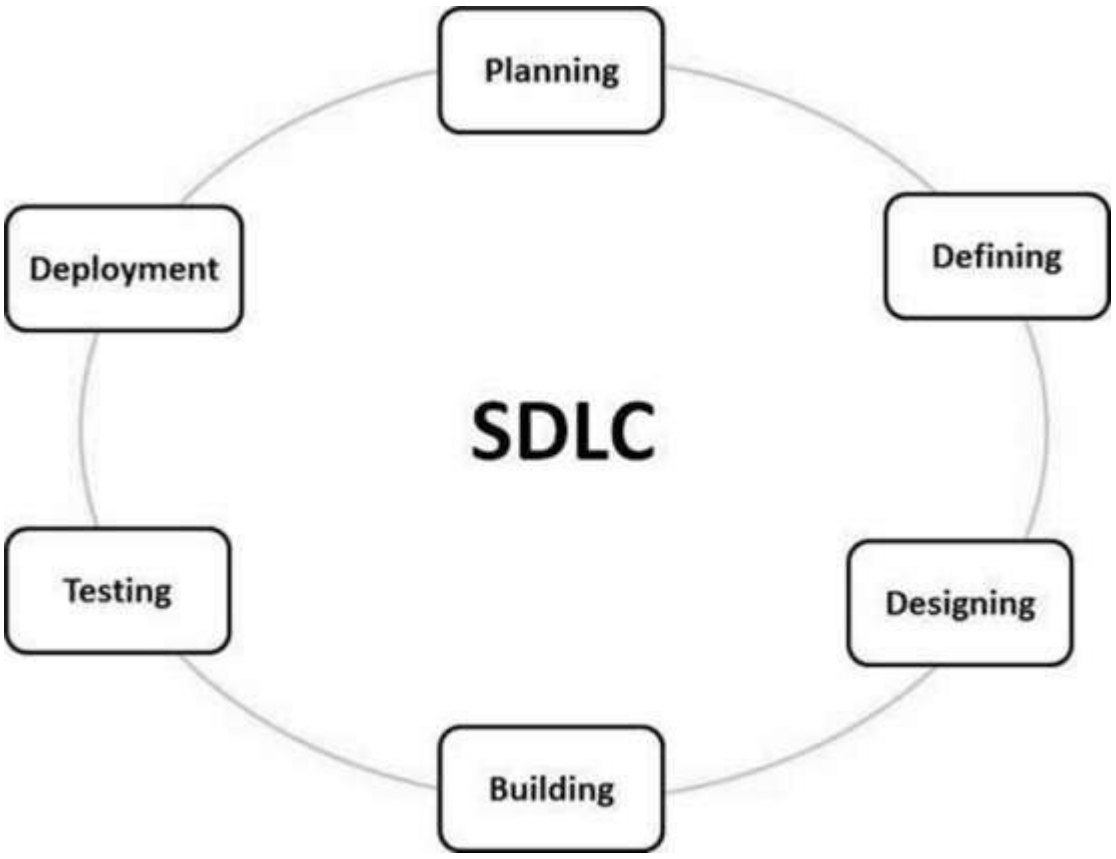
Figure 2.2: Software processes.

3.Software development process models :

What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages −

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

4.SDLC Models

There are various software development life cycle models defined and designed which are followed during the software development process. These models are also referred as Software Development Process Models". Each process model follows a Series of steps unique to its type to ensure success in the process of software development.

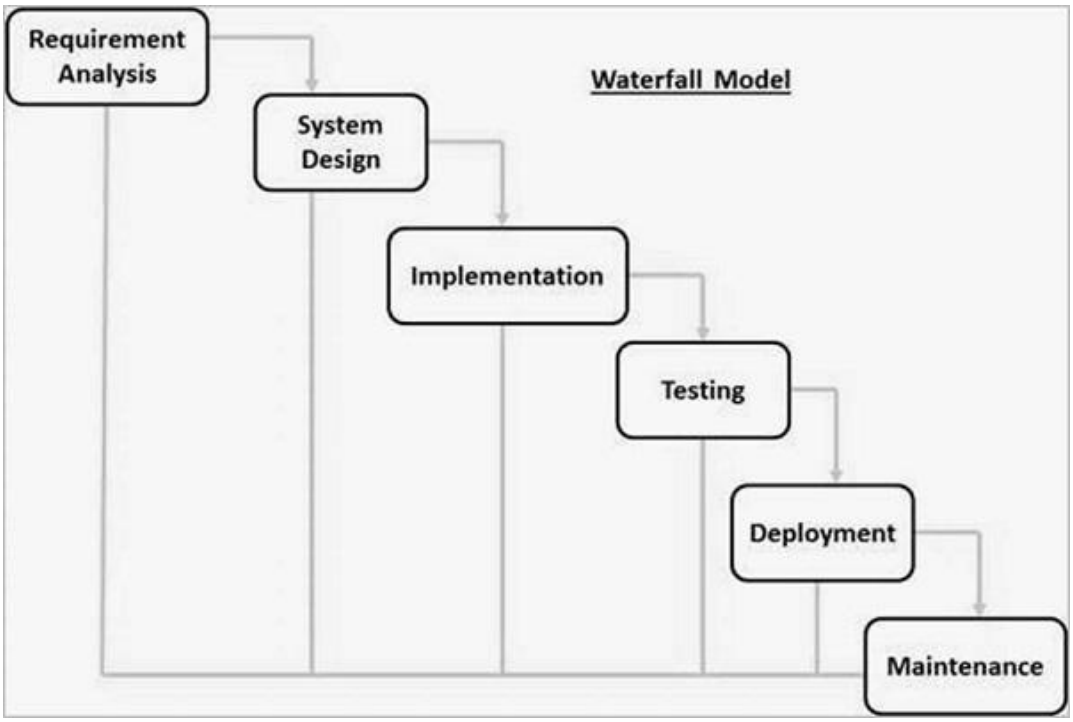
Following are the most important and popular SDLC models followed in the industry:

1. Waterfall Model
2. Prototyping
3. Iterative Model
4. Relational unified process
5. Time boxing model
6. Extreme programming
7. Agile process

1.Waterfall Model:

- Waterfall model is an example of a Sequential model. In this model, the software development activity is divided into different phases and each phase consists of series of tasks and has different objectives.
- It is divided into phases and output of one phase becomes the input of the next phase. It is mandatory for a phase to be completed before the next phase starts. In short, there is no overlapping in Waterfall model.
- In waterfall, development of one phase starts only when the previous phase is complete. Because of this nature, each phase of waterfall model is quite precise well defined. Since the phases fall from higher level to lower level, like a waterfall, It's named as waterfall model.

Pictorial representation of waterfall model:



Pros and Cons of waterfall model:

Advantages of using Waterfall model :

- Simple and easy to understand and use.
- For smaller projects, waterfall model works well and yield the appropriate results.
- Since the phases are rigid and precise, one phase is done one at a time, it is easy to maintain.

- The entry and exit criteria are well defined, so it easy and systematic to proceed with quality.
- Results are well documented.

Disadvantages of using Waterfall model:

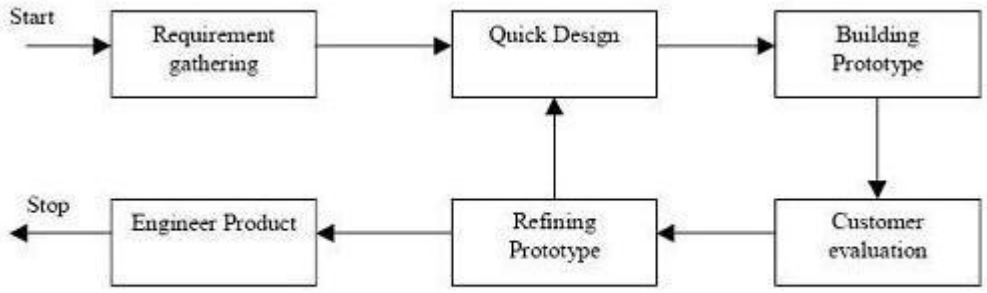
- Cannot adopt the changes in requirements
- It becomes very difficult to move back to the phase. For example, if the application has now moved to the testing stage and there is a change in requirement, It becomes difficult to go back and change it.
- Delivery of the final product is late as there is no prototype which is demonstrated intermediately.
- For bigger and complex projects, this model is not good as a risk factor is higher.
- Not suitable for the projects where requirements are changed frequently.
- Does not work for long and ongoing projects.
- Since the testing is done at a later stage, it does not allow identifying the challenges and risks in the earlier phase so the risk mitigation strategy is difficult to prepare.

Conclusion:
In the waterfall model, it is very important to take the sign off of the deliverables of each phase. As of today most of the projects are moving with Agile and Prototype models, Waterfall model still holds good for smaller projects. If requirements are straightforward and testable, Waterfall model will yield the best results.

2.Prototyping :

- The basic idea in **Prototype model** is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements.
- This prototype is developed based on the currently known requirements. Prototype model is a **software development model**. By using this prototype, the client can get an "actual feel" of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.
- Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements.
- The prototype are usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.

Diagram of Prototype model:



Prototyping Model

Advantages of Prototype model:

- Users are actively involved in the development
- Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
- Errors can be detected much earlier.
- Quicker user feedback is available leading to better solutions.
- Missing functionality can be identified easily

.

Disadvantages of Prototype model:

- Leads to implementing and then repairing way of building systems.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.

When to use Prototype model:

- Prototype model should be used when the desired system needs to have a lot of interaction with the end users.
- Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.
- Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems.

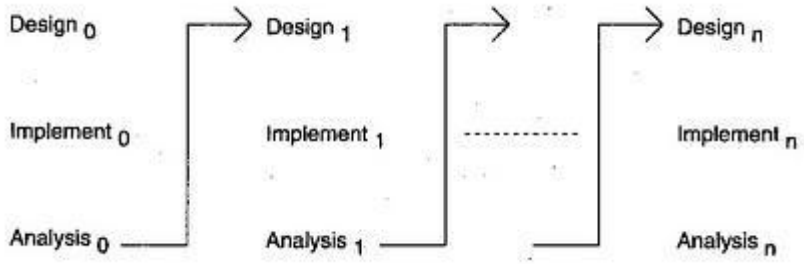
3.Iterative Model :

An iterative **life cycle model** does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

For example:

In the diagram above when we work **iteratively** we create rough product or product piece in one iteration, then review it and improve it in next iteration and so on until it's finished. As shown in the image above, in the first iteration the whole painting is sketched roughly, then in the second iteration colors are filled and in the third iteration finishing is done. Hence, in iterative model the whole product is developed step by step.

Diagram of Iterative model:



Advantages of Iterative model:

- In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
- In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In iterative model less time is spent on documenting and more time is given for designing.

Disadvantages of Iterative model:

- Each phase of an iteration is rigid with no overlaps
- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle

When to use iterative model:

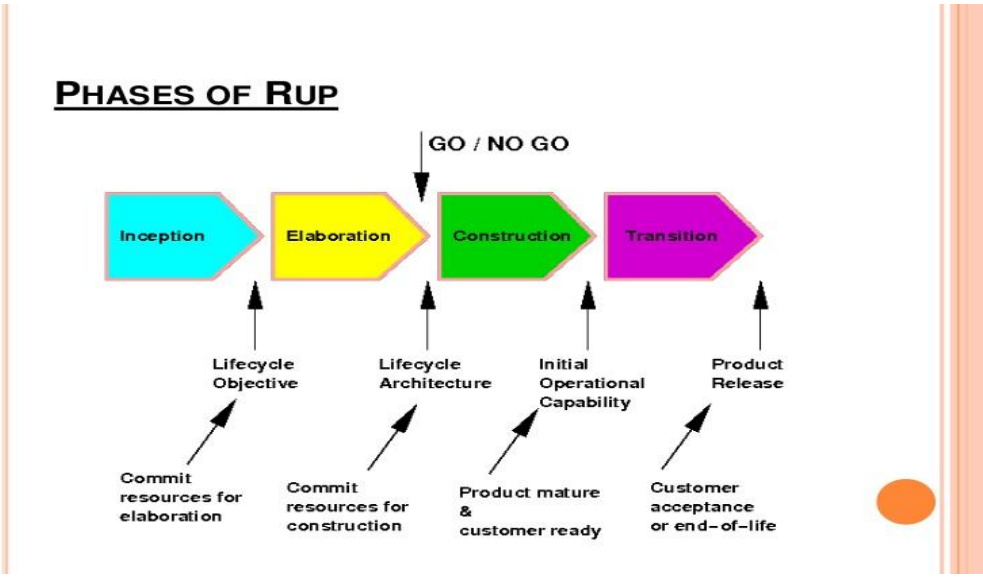
- Requirements of the complete system are clearly defined and understood.
- When the project is big.
- Major requirements must be defined; however, some details can evolve with time.

4.Relational unified process(RUP) :

RUP Stands for "Rational Unified Process." RUP is a software development process from Rational, a division of IBM. It divides the development process into four distinct phases that each involve business modeling, analysis and design, implementation, testing, and deployment. The four phases are:

1. **Inception -** The idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.
2. **Elaboration -** The project's architecture and required resources are further evaluated. Developers consider possible applications of the software and costs associated with the development.
3. **Construction -** The project is developed and completed. The software is designed, written, and tested.
4. **Transition -** The software is released to the public. Final adjustments or updates are made based on feedback from end users.

The RUP development methodology provides a structured way for companies to envision create software programs. Since it provides a specific plan for each step of the development process, it helps prevent resources from being wasted and reduces unexpected development costs.



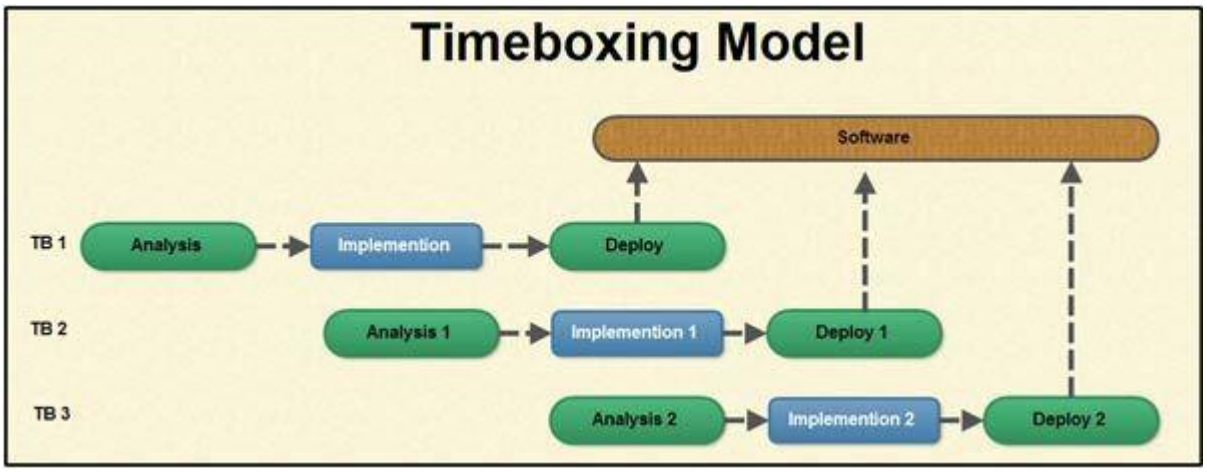
5.Time boxing model :

- In time boxing model, development is done iteratively as in the iterative enhancement model. However, in time boxing model, each iteration is done in a timebox of fixed duration. The functionality to be developed is adjusted to fit the duration of the timebox. Moreover, each timebox is divided into a sequence of fixed stages where each stage performs a clearly defined task (analysis, implementation, and deploy) that can be done independently. This model also requires that the time duration of each stage is approximately equal so that pipelining concept is employed to have the reduction in development time and product releases.
- There is a dedicated team for each stage so that the work can be done in pipelining. Thus, stages should be chosen in such a way that each stage perform some logical unit of work that becomes the input for next stage.

In addition to the advantages of iterative model, time boxing model has some other advantages too. Various advantages and disadvantages associated with timeboxing model are listed in Table.

Table Advantages and Disadvantages of the Time boxing Model

Advantages	Disadvantages
Speeds up the development process and shortens the delivery time Well suited to develop projects with a number of features in short time period.	Project management becomes more complex. Not suited to projects in which entire development work cannot be divided into multiple iterations of almost, equal duration.



6.Extreme programming

What is Extreme Programming?

XP is a lightweight, efficient, low-risk, flexible, predictable, scientific, and fun way to develop a software.

e**X**treme **P**rogramming (XP) was conceived and developed to address the specific needs of software development by small teams in the face of vague and changing requirements.

Extreme Programming is one of the Agile software development methodologies. It provides values and principles to guide the team behavior. The team is expected to self-organize. Extreme Programming provides specific core practices where −

- Each practice is simple and self-complete.
- Combination of practices produces more complex and emergent behavior.

Embrace Change

A key assumption of Extreme Programming is that the cost of changing a program can be held mostly constant over time.

This can be achieved with −

- Emphasis on continuous feedback from the customer
- Short iterations

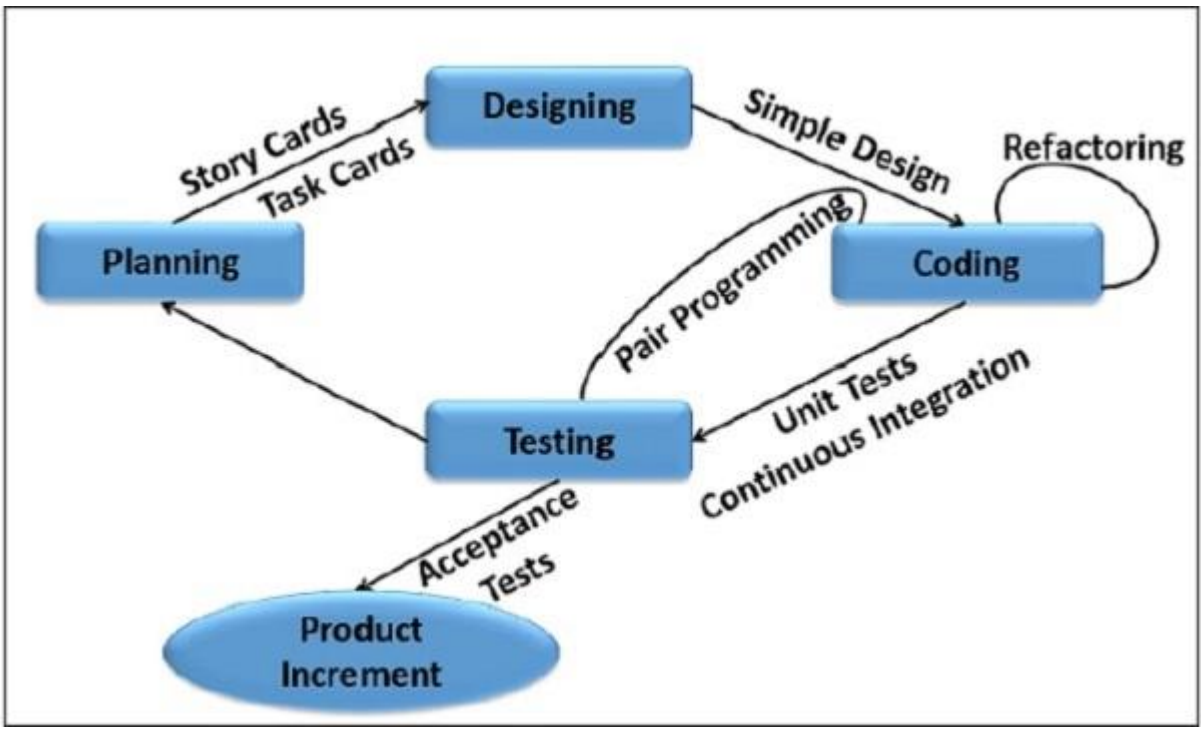
- Design and redesign
- Coding and testing frequently
- Eliminating defects early, thus reducing costs
- Keeping the customer involved throughout the development
- Delivering working product to the customer

Extreme Programming in a Nutshell

Extreme Programming involves –

- Writing unit tests before programming and keeping all of the tests running at all times. The unit tests are automated and eliminates defects early, thus reducing the costs.
- Starting with a simple design just enough to code the features at hand and redesigning when required.
- Programming in pairs (called pair programming), with two programmers at one screen, taking turns to use the keyboard. While one of them is at the keyboard, the other constantly reviews and provides inputs.
- Integrating and testing the whole system several times a day.
- Putting a minimal working system into the production quickly and upgrading it whenever required.
- Keeping the customer involved all the time and obtaining constant feedback.

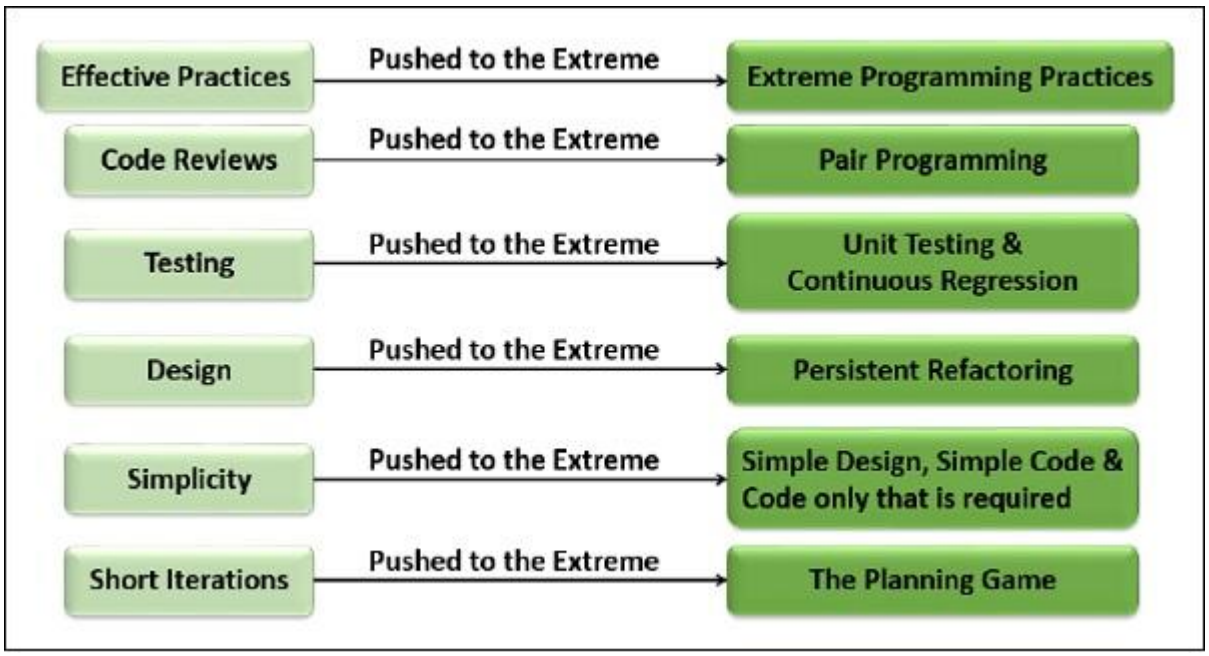
Iterating facilitates the accommodating changes as the software evolves with the changing requirements.



Why is it called "Extreme?"

Extreme Programming takes the effective principles and practices to extreme levels.

- Code reviews are effective as the code is reviewed all the time.
- Testing is effective as there is continuous regression and testing.
- Design is effective as everybody needs to do refactoring daily.
- Integration testing is important as integrate and test several times a day.
- Short iterations are effective as the planning game for release planning and iteration planning.



Success in Industry

The success of projects, which follow Extreme Programming practices, is due to –

- Rapid development.
- Immediate responsiveness to the customer's changing requirements.
- Focus on low defect rates.
- System returning constant and consistent value to the customer.
- High customer satisfaction.
- Reduced costs.
- Team cohesion and employee satisfaction.

Extreme Programming Advantages

Extreme Programming solves the following problems often faced in the software development projects –

- **Slipped schedules** – and achievable development cycles ensure timely deliveries.
- **Cancelled projects** – Focus on continuous customer involvement ensures transparency with the customer and immediate resolution of any issues.
- **Costs incurred in changes** – Extensive and ongoing testing makes sure the changes do not break the existing functionality. A running working system

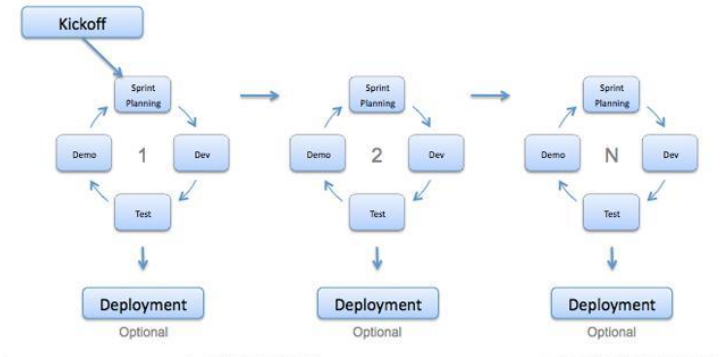
always ensures sufficient time for accommodating changes such that the current operations are not affected.

- **Production and post-delivery defects: Emphasis is on** – the unit tests to detect and fix the defects early.
- **Misunderstanding the business and/or domain** – Making the customer a part of the team ensures constant communication and clarifications.
- **Business changes** – Changes are considered to be inevitable and are accommodated at any point of time.
- **Staff turnover** – Intensive team collaboration ensures enthusiasm and good will. Cohesion of multi-disciplines fosters the team spirit.

7.Agile process Model :

Agile development model is also a type of **Incremental model**. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly **tested** to ensure **software quality** is maintained. It is used for time critical applications. Extreme Programming (XP) is currently one of the most well known agile **development life cycle model**.

Diagram of Agile model:



Advantages of Agile model:

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.

- Close, daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

Disadvantages of Agile model:

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

When to use Agile model:

- When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
- To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- Unlike the **waterfall model** in agile model very limited **planning** is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

5. Using process models in a project :

- Waterfall comes from other kinds of industries... When you see an airplane, a gigantic bridge, a 50+ floors building, all of those were made with the waterfall model (your clients just can't change your requirements midway, since you would have to demolish the building first and that costs money).

  In fact, I would dare say many projects where the size is really big and have relationships with external systems (hardware related) are built with waterfall-like models, since you can't change requirements midway without investing millions.

- Prototype model should be used when the desired system needs to have a lot of interaction with the end users. Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model.

- Agile processes can deliver successful systems quickly. It stresses on continuous communication and collaboration among developers and customers.

- *This dynamic and iterative approach to planning allows you to achieve a better balance between target -- where you want to be at the end of the project -- and commitments -- what the team must do to get there. It is adaptive, and it is driven by risks and what has been learned so far, about both the product and the process used to develop it.*

- *In the Rational Unified Process® (RUP®), planning is more focused on a process breakdown -- that is, what needs to be done to achieve certain objectives over time. You plan based on phases and iterations that include major and minor milestones.*

6. Project management process:

Project management is one of the critical processes of any project. This is due to the fact that project management is the core process that connects all other project activities and processes together.

When it comes to the activities of project management, there are plenty. However, these plenty of project management activities can be categorized into five main processes.

Let's have a look at the five main project management processes in detail.

1 - Project Initiation

Project initiation is the starting point of any project. In this process, all the activities related to winning a project takes place. Usually, the main activity of this phase is the pre-sale.

During the pre-sale period, the service provider proves the eligibility and ability of completing the project to the client and eventually wins the business. Then, it is the detailed requirements gathering which comes next.

During the requirements gathering activity, all the client requirements are gathered and analysed for implementation. In this activity, negotiations may take place to change certain requirements or remove certain requirements altogether.

Usually, project initiation process ends with requirements sign-off.

2 - Project Planning

Project planning is one of the main project management processes. If the project management team gets this step wrong, there could be heavy negative consequences during the next phases of the project.

Therefore, the project management team will have to pay detailed attention to this process of the project.

In this process, the project plan is derived in order to address the project requirements such as, requirements scope, budget and timelines. Once the project plan is derived, then the project schedule is developed.

Depending on the budget and the schedule, the resources are then allocated to the project. This phase is the most important phase when it comes to project cost and effort.

3 - Project Execution

After all paperwork is done, in this phase, the project management executes the project in order to achieve project objectives.

When it comes to execution, each member of the team carries out their own assignments within the given deadline for each activity. The detailed project schedule will be used for tracking the project progress.

During the project execution, there are many reporting activities to be done. The senior management of the company will require daily or weekly status updates on the project progress.

In addition to that, the client may also want to track the progress of the project. During the project execution, it is a must to track the effort and cost of the project in order to determine whether the project is progressing in the right direction or not.

In addition to reporting, there are multiple deliveries to be made during the project execution. Usually, project deliveries are not onetime deliveries made at the end of the project. Instead, the deliveries are scattered through out the project execution period and delivered upon agreed timelines.

4 Control and Validation

During the project life cycle, the project activities should be thoroughly controlled and validated. The controlling can be mainly done by adhering to the initial protocols such as project plan, quality assurance test plan and communication plan for the project.

Sometimes, there can be instances that are not covered by such protocols. In such cases, the project manager should use adequate and necessary measurements in order to control such situations.

Validation is a supporting activity that runs from first day to the last day of a project. Each and every activity and delivery should have its own validation criteria in order to verify the successful outcome or the successful completion.

When it comes to project deliveries and requirements, a separate team called 'quality assurance team' will assist the project team for validation and verification functions.

5 Closeout and Evaluation

Once all the project requirements are achieved, it is time to hand over the implemented system and closeout the project. If the project deliveries are in par with the acceptance criteria defined by the client, the project will be duly accepted and paid by the customer.

Once the project closeout takes place, it is time to evaluate the entire project. In this evaluation, the mistakes made by the project team will be identified and will take necessary steps to avoid them in the future projects.

During the project evaluation process, the service provider may notice that they haven't gained the expected margins for the project and may have exceeded the timelines planned at the beginning.

In such cases, the project is not a 100% success to the service provider. Therefore, such instances should be studied carefully and should take necessary actions to avoid in the future.

Conclusion

Project management is a responsible process. The project management process connects all other project activities together and creates the harmony in the project.

Therefore, the project management team should have a detailed understanding on all the project management processes and the tools that they can make use for each project management process.

UNIT 2

[QUESTIONS]

1. What is software process and project.
2. Explain about component software process.
3. What is SDLC. What are different SDLC models.
4. Describe classical waterfall model and iterative development model of Software development. Draw appropriate diagrams. Compare the two models.
5. Discuss about prototyping model. Explain its merits and Demerits.
6. Discuss in detail about the layers involved in Unified Process model.
7. Discuss in detail about Time Boxing model.
8. Explain different phases of Extreme Programming (XP) with a neat diagram .

[or]

What are the phases of extreme programming? Explain.

[or]

Describe extreme programming process.

9. What is Agility? What is an agile process?
10. What is Agility? Explain Extreme programming Values.

11. What are generic process models and explain process models?[Note : Generic process Model = WaterFall Model and explain all other models for process models.]

12. Explain about project management process.

13. What are perspective process models.

[There are two types of prescriptive process models. They are:
1. The Waterfall Model
2. Incremental Process model]

NOTE: when question is asked like this explain about waterfall and incremental.]

14. Explain about agile model for software development. Compare it with waterfall model, prototyping model and iterative development model.

15.) Why does an iterative process make it easier to manage change? Is it possible to complete a project in just one iteration and still be agile? Explain your answers.

16.)Describe how incremental process models are better than water fall model.

17.) What process models will you use in various projects.

[**NOTE :** Write this answer from your notes (Using process models in a project)]