# **Chapter 1: Software Engineering Process**

## **Software Process Lifecycle**

A key factor behind every successful software development project is to establish a process for development. Process organizes the development of a software product into multiple distinct phases or stages, for example, Specification, Design and Implementation, Verification and Validation phases. A process can be the entire life cycle of a software product or can be a sub-process within a life cycle process. Each phase is composed of associated activities. An activity is a group of related tasks (low-level atomic actions). Tasks are the small, manageable units of work for the project. They are the building blocks for completing an activity, finishing a phase, and ultimately completing a process [1].

Every taskrelates to roles, work products, and resources.A role is job-related activity assigned to a person (i.e. a role performs a task). Each role on a team describes specific skills that are needed to accomplish any given responsibility or task. For example, a programmer may write code for a certain feature. And a tester will be assigned to test that feature, and this role may be filled by a Junior Developer. A work product is an output produced by completing a specific task. Work productsinclude the final product and they also include other intermediate outputs such as design documentation, requirements documents, source code, test cases, and project management artifacts such as status reports and sign-off forms. Since work products are outputs of tasks, we can say that a task produces a work product.Not only does a task produce work products as output, it also uses work products from another task as input. Completing any given task requires resources to advance or fund it, such as time, people, technology, knowledge, and money, equipment, etc. [1].





Practices are strategies used to execute processes smoothly. Practices contribute to the effectiveness of a development process. For instance, a manager can follow proven management practices for: scheduling and organizing tasks, minimizing resource waste, and tracking completed work. Additionally, estimating tasks, holding daily meetings, improving communication, and doing periodic reviews of the project are all practices that can help a process run well. A developer can follow established development practices for: preparing effective tests, maintaining simplicity, and conducting technical reviews. This ensures a quality product with low defects.Practices are often gathered into a methodology i.e. a methodology is a set of practices. Methodologies based on the philosophy outlined in the Agile Manifesto are known as Agile methodologies. Scrum is an example of an Agile methodology[1].

Adopting good practices reduces wasted resources. Effective planning and communication prevent duplicating work or producing a feature that is not needed. Good practices then help to keep the project on schedule and on budget. Processes and practices are flexible and customizable. Combine various aspects of processes and practices to create something customized to your development needs. Finally, be open to making adjustments that refine the processes and practices to best fit your project and product [1].

Creating software is more like authoring a cookbook than following a specific recipe. There is no prescribed formula to writing a cookbook; recipes must be collected, organized, and tested. Recipe testing may lead to further refinements, experiments, and improvisations. Rejection of some recipes may require backtracking to the collection phase. And, if successful, a cookbook may have a long-life cycle involving multiple editions in which corrections are made, and new recipes are added. Similarly, software development requires continuous evaluation and decision making while moving through each phase of a given project. This continues well past the initial development project. A software product, just as with a cookbook, can have a long-life cycle involving multiple versions with improved features and resolved defects[1].

## **EXERCISE -1: Problem Analysis**

**Objective:**Develop a Project Proposal

**Tools/ Device:**Using Mind Mapping tools can be very useful to express your ideas and discussion with other group members. Here are some useful online mind mapping tools

1. Coggle: <https://coggle.it>
2. Mindmester: <https://www.mindmeister.com>
3. Lucidchart (a business process design modeling tool): <https://www.lucidchart.com>

**Procedure:**

1. **Self-Study:** open an appropriate software engineering guide and study the software development life cycle and related topics.Study the needs of the software engineering.
2. **Formulating Team:** A team is up to four students. In addition to each team, other people are allowed to help. A team can have a mentor, who advises on the project. A team can also have associates, who must be eligible students that can help the team with things, with business model planning, marketing, and so on. These people are not members of the team as such but it's good to have help sometimes.
3. **Identify Problem Domain:**Each team is required to come up with an idea of software development (either Problem domain: *Category A* or *Category B*). It can be either purely software or a combination of software and hardware. You must choose at least one of the options for *technology*(Web, Desktop, Mobile Phone/Handheld Devices), but if any team is optimistic enough: can choose all of the available options.

**Category A:**Find a real-life problem, even in your own life or community, and then work to solve it. Build a project that could change lives. The next big thing could come from you. Facebook and Twitter started as student projects. Your ideas could be next. Come up with an innovative application idea and proceed accordingly.

**Category B:**There might be dozens of software to solve our real-life problems and provide benefit to business. Come up with an idea that will extend the current version of those existingsoftware.

1. **Prepare a Project Proposal:** write the background description that helps putting your project into the right context of a problem domain and gives everyone involved a common view of the project. What is the root cause of this problem? why is this problem is so important to consider?What is your project objective? What solution are you going to provide to solve the above-mentioned problem. Who are the target group of users of your solution? And how they will be benefited by your proposed solution to the problem? What are the basic functionalities of your proposed solution?

**Evaluation:**Your Project Proposal must address the following questions:

1. Does the team demonstrate a thorough understanding of the need, problem, oropportunity, including evidence of research into the need, problem, or opportunity?
2. Does the project have a clear objective including relevant benefits and target market oraudience of the product?
3. What are the solutions you are going to propose to deal with the problem? why is this   
   solution is particularly appropriate to solve the problem? Is the solution feasible to themeet the business objective?
4. Is the project’s purpose and basic functionality easily understood?