Continuation of Lecture 2

Legacy Software

What is Legacy Software?

- For centuries, people used the term “legacy” to refer to items of value handed down from generation to generation.
- However, in the IT world, the **term carries a negative connotation, conjuring up images of outdated, underperforming systems that struggle to retain their relevance in a digital-first marketplace**.
- In this era of accelerated growth, organizations are discovering that their once efficient systems are now unable to meet their business objectives.

Legacy systems

- Legacy systems are defined as those that rely on older or obsolete technology.
- This includes not only **outdated computer hardware** but also **programming languages and software applications** that can no longer support an organization’s technology needs.

Legacy Software

- Legacy software systems are also responsible for costly delays and disruptions.
- Maintaining legacy software can expose organizations to data breaches due to outdated security measures, performance issues, and non-compliance to security standards.
- **To determine exactly what legacy software is, consider the following conditions:**
    - Older software supported with feature updates but no security patches
    - A platform incompatible with new systems and drivers
    - Software that’s non-compliant with recent standards
    - Software that requires new updates to be functional

Examples of Legacy Software

- **Microsoft Windows 7:** Windows 7 officially became a legacy operating system in January 2020 after Microsoft halted security updates and support for it. However, over 100 million machines continue to run this operating system.
- **COBOL:** Common Business-Oriented Language or COBOL is still used 55 years after its development. Forty-eight percent of businesses and government organizations reportedly depend on this language more than others.
- **Discontinued Oracle products:** Oracle database software such as E-Business Suite and Peoplesoft.

Software process, Software project, Software product

- A software process, specifies a method of development software.
- A software project, on the other hand is a development project in which a software process is used.
- And software products are the outcomes of a software project.
- Each software development **project starts with some needs** and (hopefully) **ends with some software that satisfies those needs**.
- A **software process** specifies the abstract set of activities that should be performed to go from user needs to final product.
- The actual act of executing the activities for some specific user needs is a **software project**.
- And all the outputs that are produced while the activities are being executed are the **products**.

6 Process Categories in Software Engineering

- Software Life Cycle Model Process
- Project Management Processes
- Pre-Development Processes
- Development Processes
- Post-Development Processes
- Integral Processes

- The selection of the **development process decides the phases and tasks to be done**
- It **does not specify things like**
  - **how long each phase should last, or**
  - **how many resources should be assigned to a phase, or**
  - **how a phase should be monitored**.
- Quality and productivity in the project also depend critically on these decisions.

- **To meet the cost, quality, and schedule objectives:**
  - **resources** have to be **properly allocated** to each activity for the project, and
  - **progress** of different activities has to be **monitored** and
  - **corrective actions** taken when needed.
- **All these activities are part of the project management process.**

Project Management Process

- The 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**.
  - That is, the basic task is to **plan the detailed implementation of the process** for the particular project and then **ensure that the plan is properly executed**.
- For a large project, a proper management process is essential for success.

Process Management Definitions

- Process Management is the discipline of defining, implementing and maintaining the work processes within an organization.
- The goal of Process Management is to create an environment for improving quality and productivity.
- A foundation of a successful Process Management system is a defined framework that fits the organization's objective and culture.
- Building a Process Management system is a progressive iterative task that requires a strategic commitment of the organization.

Why is the Process Management important?

- “The **quality of a product** (e.g., a software system) is governed by the **quality of the process** used to produce it.”

Adapted from Book “Quality Software Project Management” Chapter 3, by Robert T. Futrell

Why is Quality Important?

- 99.9% - is it “GOOD ENOUGH”?
- If 99.9% Were “Good Enough”
  - 16,000 pieces of mail would be lost each hour.
  - We would experience 1 hour of unsafe drinking water each month.
  - 20,000 incorrect drug prescriptions would be filled each year.
  - 22,000 checks would be deducted from the wrong bank accounts each hour.
  - 50 newborn babies would be dropped at birth by their doctors every day.
  - 500 incorrect surgical operations would be performed each week.

Stages of Project Management Process

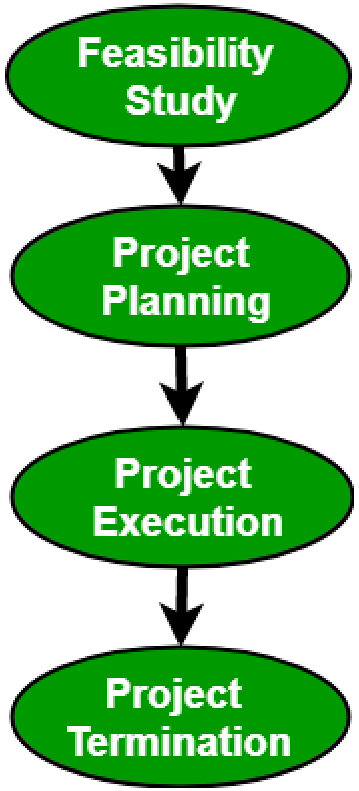
Project Management Process consists of the following 4 stages:

- **Feasibility study**
- **Project Planning**
- **Project Execution**
- **Project Termination**

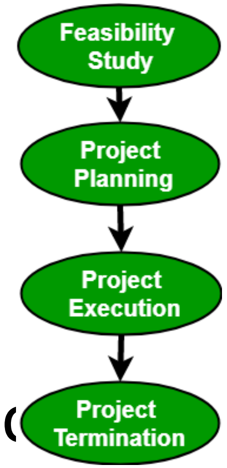
- NOTE: Some authors have grouped the management process for a project, broadly into three phases: planning, monitoring and control, and termination analysis.

Stages of Project Management Process

Project Management Process consists of the following 4 stages:



Feasibility Study:

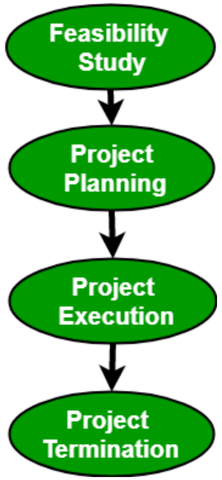


- A feasibility study explores **system requirements** to **determine project feasibility**.
- There are several fields of a feasibility study including:
  - economic feasibility,
  - operational feasibility,
  - technical feasibility.
- The **goal** is to determine whether the **system can be implemented or not.**

Feasibility Study: Input and Output

- The process of feasibility study takes as **input** the required details as specified by the user (Requirement Engineering) and other domain-specific details.
- The **output** of this process simply tells whether the project should be undertaken or not and **if yes**,
  - What would the **constraints** be.
  - Additionally, all the **risks and their potential effects** on the projects are also **evaluated** before a decision to start the project is taken.

Project Planning:

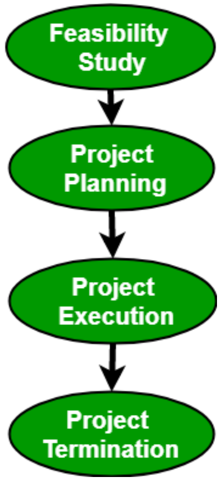


- A detailed plan stating a stepwise strategy to achieve the listed objectives is an integral part of any project.
- Planning consists of the following activities:
  - Set objectives or goals
  - Develop strategies
  - Develop project policies
  - Determine courses of action
  - Making planning decisions
  - Set procedures and rules for the project
  - Develop a software project plan
  - Prepare budget
  - Conduct risk management
  - Document software project plans

Project Planning:

- This step also involves the construction of a **work breakdown structure**(WBS).
  - **A work breakdown structure (WBS) is a visual, hierarchical and deliverable-oriented deconstruction of a project.**
  - It is a helpful diagram for project managers because it allows them to break down their project scope and visualize all the tasks required to complete their projects.
- Project planning also includes size, effort, schedule, and cost estimation using various techniques.

Project Execution:

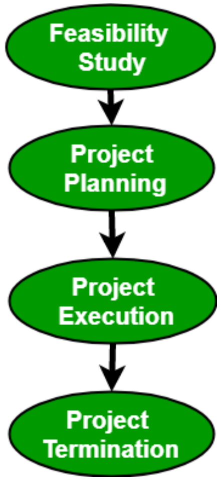


- A project is executed by **choosing an appropriate software development lifecycle model(SDLC)**.
- It includes a number of steps including
  - **Requirements analysis,**
  - **Design Engineering,**
  - **Coding and Testing**
  - **Implementation,**
  - **Delivery, Testing – Reliability and Quality, and Maintenance.**

Project Execution:

- There are a number of factors that need to be considered while choosing an appropriate SDLC, including:
  - the size of the system,
  - the nature of the project,
  - time and budget constraints,
  - domain requirements, etc.
- An inappropriate SDLC can lead to the failure of the project.

Project Termination:



- There can be several reasons for the termination of a project.
- Though expecting a project to **terminate after successful completion** is conventional, at times, a **project may also terminate without completion**.
- Projects have to be closed down when the **requirements are not fulfilled according to given time and cost constraints**.

Project Termination:

- Some of the reasons for failure include:
    - Fast-changing technology
    - Project running out of time
    - Organizational politics
    - Too much change in customer requirements
    - Project exceeding budget or funds
- Once the project is terminated, a **post-performance analysis** is done.
- **Also, a final report is published describing the experiences, lessons learned, recommendations for handling future projects.**