Software Project Management

Engr. Madeha Mushtaq
Department of Computer Science
Iqra National University

Project Characteristics

- Following are the project characteristics:
- Temporary:
    - Temporary means that every project has a definite beginning and a definite end.
    - Temporary does not necessarily mean short in duration; many projects last for several years.
    - The duration of a project is finite; projects are not ongoing efforts.
- Unique, Product Service or Result:
    - Projects involve creating something that has not been done before and which is, therefore, unique and distinct.

Project Characteristics

  - Projects create:
    - A product or artifact that is produced and is quantifiable.
    - A capability to perform a service, such as business functions supporting production or distribution.
    - A result, such as new knowledge. For example, a research and development project.
- Aims/Tasks/Purpose:
  - The projects are designed to achieve specific targets defined in terms of aims, tasks or a purpose.
  - A project thus has to be aimed for achieving certain tasks in a given time frame.

Project Characteristics

- Limited Time Scale:
  - The projects are always designed considering time constraints.
  - Extension to the project completion deadline are always discouraged.

Four Project Dimensions

- Effective software project management focuses on the four P’s: people, product, process, and project.
- **People** must be organized into effective teams, motivated to do high-quality software work, and coordinated to achieve effective communication.
- The **product** requirements must be communicated from customer to developer, partitioned (decomposed) into their constituent parts, and positioned for work by the software team.

Four Project Dimensions

- The **process** must be adapted to the people and the problem.
- A common process framework is selected and an appropriate software engineering paradigm is applied.
- The **project** must be organized in a manner that enables the software team to succeed.

Important Players

- The software process (and every software project) is populated by players who can be categorized into one of the following:
- **Senior managers** who define the business issues that often have significant influence on the project.
- **Project (technical) managers** who must plan, motivate, organize, and control the practitioners who do software work.
- **Practitioners** who deliver the technical skills that are necessary to engineer a product or application.
- **Customers** who specify the requirements for the software to be engineered and other stakeholders who have a peripheral interest in the outcome.

Important Players

- End-users:
  - Who interact with the software once it is released for production use.
- Team Leaders:
  - Project management is a people-intensive activity.
  - The “people factor’ is so important that the Software Engineering Institute has developed a people management capability maturity model (PM CMM), “to enhance the readiness of software organizations to undertake increasingly complex applications.
  - The people management maturity model defines the following key practice areas for software people: recruiting, selection, performance management, training, compensation, career development, organization and work design.

Project Phases

- Organizations performing projects will usually divide each project into several Project phases to improve management control.
- Collectively, the project phases are known as the project life cycle.
- This view of software development is referred to as the software development life cycle.

Project Phases

- Every project has five phases:
- Initiation:
    - Articulate your vision for the project, establish goals, assemble your team, and define expectations and the scope of your project.
- Planning:
    - Refine the scope, identify specific tasks and activities to be completed, and develop a schedule and budget.
- Executing:
    - Accomplish your goals by leading your team, solving problems, and building your project.

Project Phases

- Controlling:
  - Monitor changes to the project make corrections, adjust your schedule to respond to problems, or adjust your expectations and goals.
- Closing:
  - Deliver your project to your audience, acknowledge results, and assess its success.
  - Take the time to compose a written evaluation of the project and the development effort.

Project Phases

- The middle three phases are not sequential.
- We will find that we are constantly planning, executing, and controlling our project as necessary.
- Aren't these phases really just common sense?
- In many ways, yes, but keep in mind that software development is a complex, unpredictable process.
- The more effort we put into managing our project, the more we increase our chances of success.

Characteristics of Project Phases

- Each project phase is marked by completion of one or more deliverables.
- A deliverable is a tangible, verifiable work product such as a feasibility study, a detail design, or a working prototype.
- The conclusion of a project phase is generally marked by a review of both key deliverables and project performance to date.
- These phase-end reviews are often called phase exits, stage gates, or kill points.

Characteristics of the Project Life Cycle

- The project life cycle serves to define the beginning and the end of a project.
- The project life-cycle definition will also determine which transitional actions at the beginning and the end of the project are included and which are not.
- In this manner, the project life-cycle definition can be used to link the project to the ongoing operations of the performing organization.
- The phase sequence defined by most project life cycles generally involves some form of technology transfer or handoff such as requirements to design, construction to operations, or design to manufacturing.
- Deliverables from the preceding phase are usually approved before work starts on the next phase.

Characteristics of the Project Life Cycle

- Sometimes the project phases overlap.
- This practice of overlapping phases is often called fast tracking.
- Project life cycles generally define:
- What technical work should be done in each phase
- Who should be involved in each phase
- Project life-cycle descriptions may be very general or very detailed.
- Most project life-cycle descriptions share a number of common characteristics:

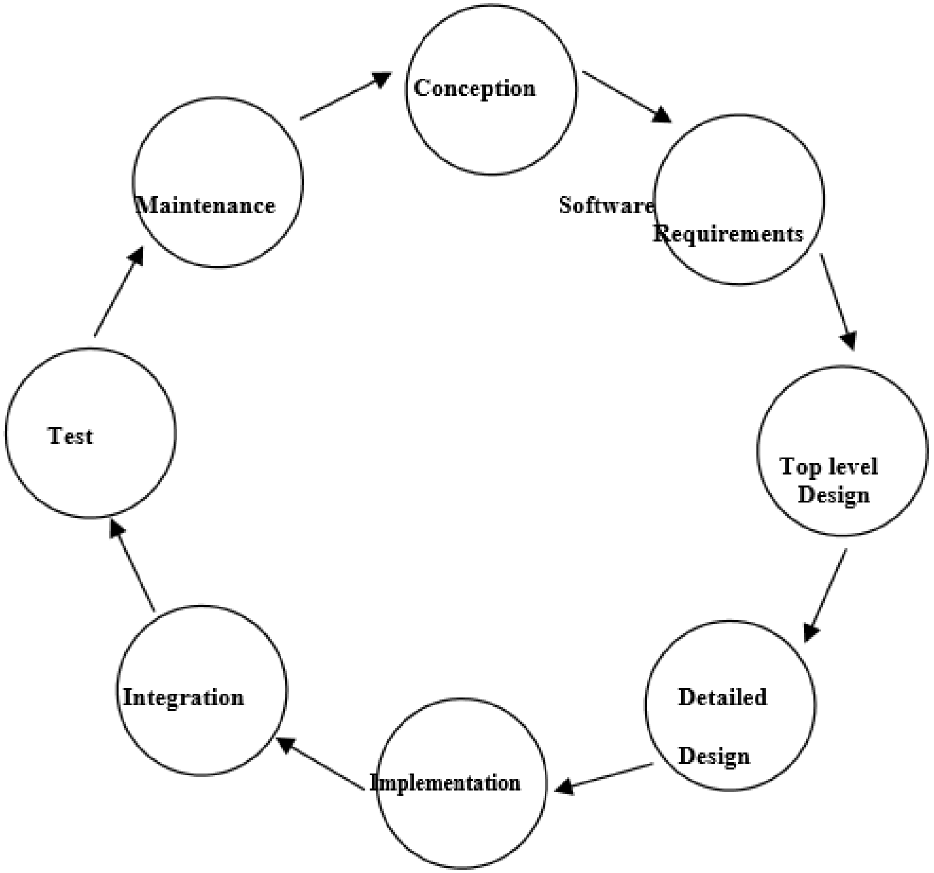
Characteristics of the Project Life Cycle

- Cost and staffing levels are low at the start, higher toward the end, and drop rapidly as the project draws to a conclusion.
- The probability of successfully completing the project is lowest, and hence risk and uncertainty are highest, at the start of the project. The probability of successful completion generally gets progressively higher as the project continues.
- The ability of the stakeholders to influence the final characteristics of the project’s product and the final cost of the project is highest at the start and gets progressively lower as the project continues. A major contributor to this phenomenon is that the cost of changes and error correction generally increases as the project continues.

Characteristics of the Project Life Cycle

- One organization’s software development life cycle may have a single design phase while another’s has separate phases for functional and detail design.
- Subprojects within projects may also have distinct project life cycles.

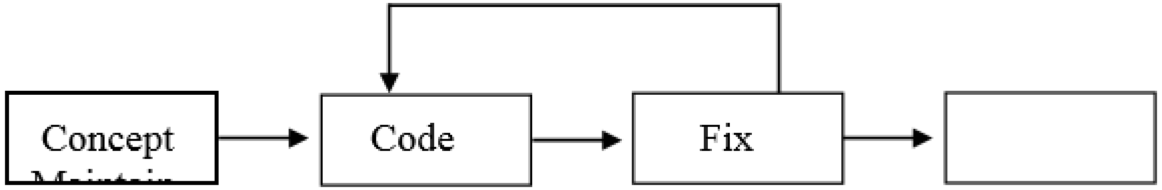
Software Development Lifecycle



The phased model of the software development life cycle

Software Development Lifecycle

- There are many variations of the software development life cycle.
- Code and Fix :
- Figure below presents a simple life cycle that was common during the first few decades of software development.

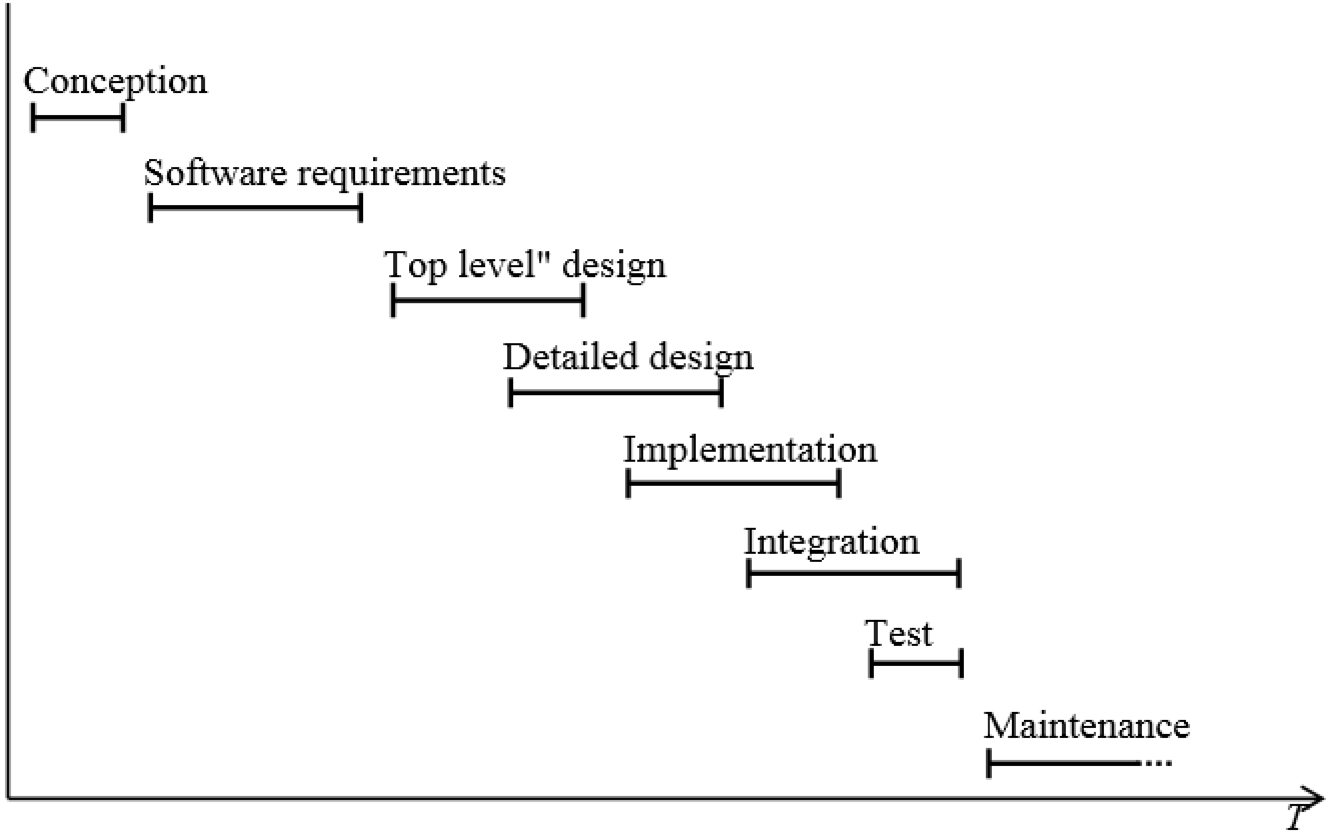


Software Development Lifecycle

- In those early days of software development, the programmer would create programs by iterating from code to fix then back to code, and then to fix again.
- At the start of the cycle, there was usually no clear concept of what was required, and the basic development procedure was a form of 'let's see what we can do' approach.

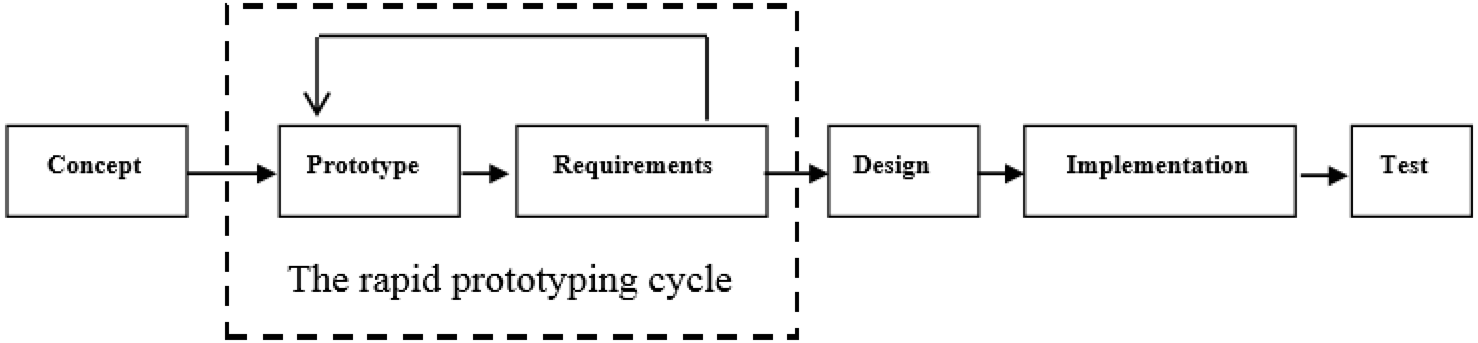
Software Development Lifecycle

- Waterfall model:
- The Waterfall model, gets its name from the way in which each phase cascades into the next (due to overlapping), as demonstrated in Figure:



Software Development Lifecycle

- Rapid prototyping:
- There are other development methodologies that do not move from one phase to the next like the Waterfall model.
- Rapid prototyping, for instance, iterates in a mini-development phase until a system prototype is developed as shown in figure:



Software Development Lifecycle

- After the prototype is complete, the Waterfall approach can then be implemented to complete the full system.
- Rapid prototyping is particularly helpful in projects where the requirements are difficult to specify.
- The prototype can be used as a tool for analyzing and determining what the requirements should be.

Software Development Lifecycle

- The Spiral model:
- Described by Boehm (1988), is another development method that iterates between the requirements, design and implementation phases.
- However, the Spiral model continues iterating until the final system is complete.
- Within each, iteration, the Spiral model follows a phased approach similar to the Waterfall model.

End of Slides