Chapter 2

Software Development Life Cycle and Process Models

Declaration

- These slides are made for UIT, BU students only. I am not holding any copy write of it as I had collected these study materials from different books and websites etc. I have not mentioned those to avoid complexity.

Topics as per Syllabus

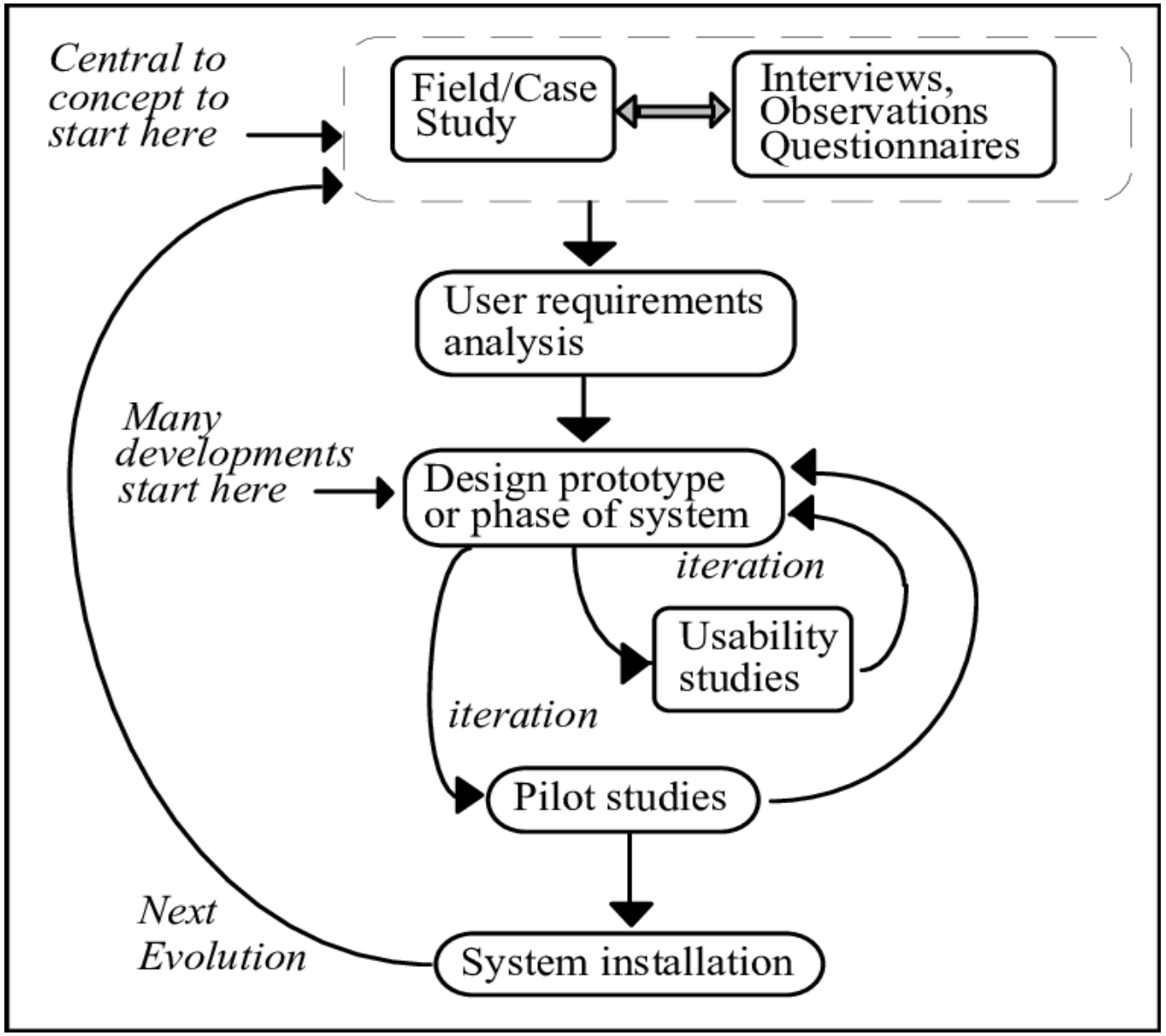
Software Development Life Cycle and Process Models:

Requirement analysis, Software Design, Coding, Testing, Maintenance. Code and Fix Model, Waterfall Model, Prototyping model, Iterative Enhancement Model, RAD Model, Evolutionary process Model, Unified process Model, Spiral Model, Selection of Life Cycle Models, Role of Management in Software Development.

Evolutionary Process Models

- **It** is a combination of Iterative and Incremental model of software development life cycle.
- It is better for software products that have their feature sets redefined during development because of user feedback and other factors. The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle.
- Evolutionary model suggests breaking down of work into smaller chunks, prioritizing them and then delivering those chunks to the customer one by one. The number of chunks is huge and is the number of deliveries made to the customer. The main advantage is that the customer's confidence increases as he constantly gets quantifiable goods or services from the beginning of the project to verify and validate his requirements. The model allows for changing requirements as well as all work in broken down into maintainable work chunks.

Evolutionary Process Models



Evolutionary Process Models-Application

- It is used in large projects where you can easily find modules for incremental implementation. Evolutionary model is commonly used when the customer wants to start using the core features instead of waiting for the full software.
- Evolutionary model is also used in object oriented software development because the system can be easily portioned into units in terms of objects.

Evolutionary Process Models- Advantages

- In evolutionary model, a user gets a chance to experiment partially developed system.
- It reduces the error because the core modules get tested thoroughly.

Evolutionary Process Models- Disadvantages

- Sometimes it is hard to divide the problem into several versions that would be acceptable to the customer which can be incrementally implemented and delivered.

Rational Unified Process

The fundamental purpose of the Rational Unified Process is to provide a model for effectively implementing commercially proven approaches to development, for use throughout the entire software development life cycle. Taking elements from other iterative software development models, the Rational Unified Process framework was initially created by the Rational Software Corporation, which was bought out by IBM in 2003.

The Rational Unified Process Best Practices

- The Rational Unified Process is structured around six fundamental best practices
  - **Develop Software Iteratively**: Encourages iterative development by locating and working on the high-risk elements within every phase of the software development life cycle.
  - **Manage Requirements**: Describes how to organize and keep track of functionality requirements, documentation, tradeoffs and decisions, and business requirements.
  - **Use Component-Based Architectures**: Emphasizes development that focuses on software *components* which are reusable through this project and, most importantly, within future projects.

The Rational Unified Process Best Practices

- **Visually Model Software**: Based on the Unified Modeling Language (UML), the Rational Unified Process provides the means to visually model software, including the components and their relationships with one another.
- **Verify Software Quality**: Assists with design, implementation, and evaluation of all manner of tests throughout the software development life cycle.
- **Control Changes to Software**: Describes how to track and manage all forms of change that will inevitably occur throughout development, in order to produce successful iterations from one build to the next.

The Building Blocks

- All aspects of the Rational Unified Process are based on a set of building blocks, which are used to describe **what** should be produced, **who** is in charge of producing it, **how** production will take place, and **when** production is complete.
- Workflows (When) are further divided up in the Rational Unified Process into six core engineering workflows:
  - **Business Modeling Workflow**
  - **Requirements Workflow**
  - **Analysis & Design Workflow**
  - **Implementation Workflow**
  - **Test Workflow**
  - **Deployment Workflow**

The Four Life Cycle Phases

- Inception Phase
  - During the inception phase, the basic idea and structure of the project is determined. The team will sit down and determine if the project is worth pursuing at all, based on the proposed purpose of the project, the estimated costs (monetary and time), and what resources will be required to complete the project once the green signal is given.
- Elaboration Phase
  - The purpose of the elaboration phase is to analyze the requirements and necessary architecture of the system. The success of this phase is particularly critical, as the final milestone of this phase signifies the transition of the project from low-risk to high-risk, since the actual development and coding will take place in the following phase.

The Four Life Cycle Phases

- Construction Phase
  - As the meat and potatoes of the software development life cycle, the construction phase is when the coding and implementation of all application features will take place. This period is also where integrations with other services or existing software should occur.
- Transition Phase
  - Easier thought of as deployment, the transition phase is when the finished product is finally released and delivered to customers. However, the transition phase is more than just the process of deployment; it must also handle all post-release support, bug fixes, patches, and so forth.
- Iterations
- The Rational Unified Process also recommends that each of the four above phases be further broken down into iterations

Rational Unified Process - Advantages

- Allows for the adaptive capability to deal with changing requirements throughout the development life cycle, whether they be from customers or from within the project itself.
- Emphasizes the need (and proper implementation of) accurate documentation.
- Diffuses potential integration headaches by forcing integration to occur throughout development, specifically within the construction phase where all other coding and development is taking place.

Rational Unified Process- Disadvantages

- Heavily relies on proficient and expert team members, since assignment of activities to individual workers should produce tangible, pre-planned results in the form of artifacts.
- Given the emphasis on integration throughout the development process, this can also be detrimental during testing or other phases, where integrations are conflicting and getting in the way of other, more fundamental activities.
- Arguably, Rational Unified Process is a fairly complicated model. Given the assortment of the components involved, including best practices, phases, building blocks, milestone criteria, iterations, and workflows, often proper implementation and use of the Rational Unified Process can be challenging for many organizations, particularly for smaller teams or projects.

Role of Management in Software Development.

Management is very important whenever we work on anything, especially in cases when we are working in a team and the number of co-workers is huge. If we talk specifically about the software development process, then the main aim of software engineering is to define a procedure which is applicable to all the software that needs to be developed, and through which we can successfully finish our project till its deployment stage and also the final product that we get is an efficient one. In short, software engineering somewhere directly or indirectly deals with the management part of the software development too.

Role of Management in Software Development - Factors

- **People**
  - Of course, the management has to deal with people in every stage of the software developing process. From the ideation phase to the final deployment phase, including the development and testing phases in between, there are people involved in everything, whether they be the customers or the developers, the designers or the salesmen. Hence, how they contact and communicate with each other must be managed so that all the required information is successfully delivered to the relevant person and hence there is no communication gap between the customers and the service providers.

Role of Management in Software Development- Factors

- **Project**
  - From the ideation phase to the deployment phase, we term the process as a project. Many people work together on a project to build a final product that can be delivered to the customer as per their needs or demands. So, the entire process that goes on while working on the project must be managed properly so that we can get a worthy result after completing the project and also so that the project can be completed on time without any delay.
- **Process**
  - Every process that takes place while developing the software, or we can say while working on the project must be managed properly and separately. For example, there are various phases in a software development process and every phase has its process like the designing process is different from the coding process, and similarly, the coding process is different from the testing. Hence, each process is managed according to its needs and each needs to be taken special care of.

Role of Management in Software Development- Factors

- **Product**
  - Even after the development process is completed and we reach our final product, still, it needs to be delivered to its customers. Hence the entire process needs a separate management team like the sales department.

End of Chapter 2

Questions?