CSE4708: Software Project Management

Unit II : Project Evaluation & Estimation

Topic: Software Process Models : The Waterfall Model, Iterative Enhancement Model, Rapid Application Development, Prototyping Models

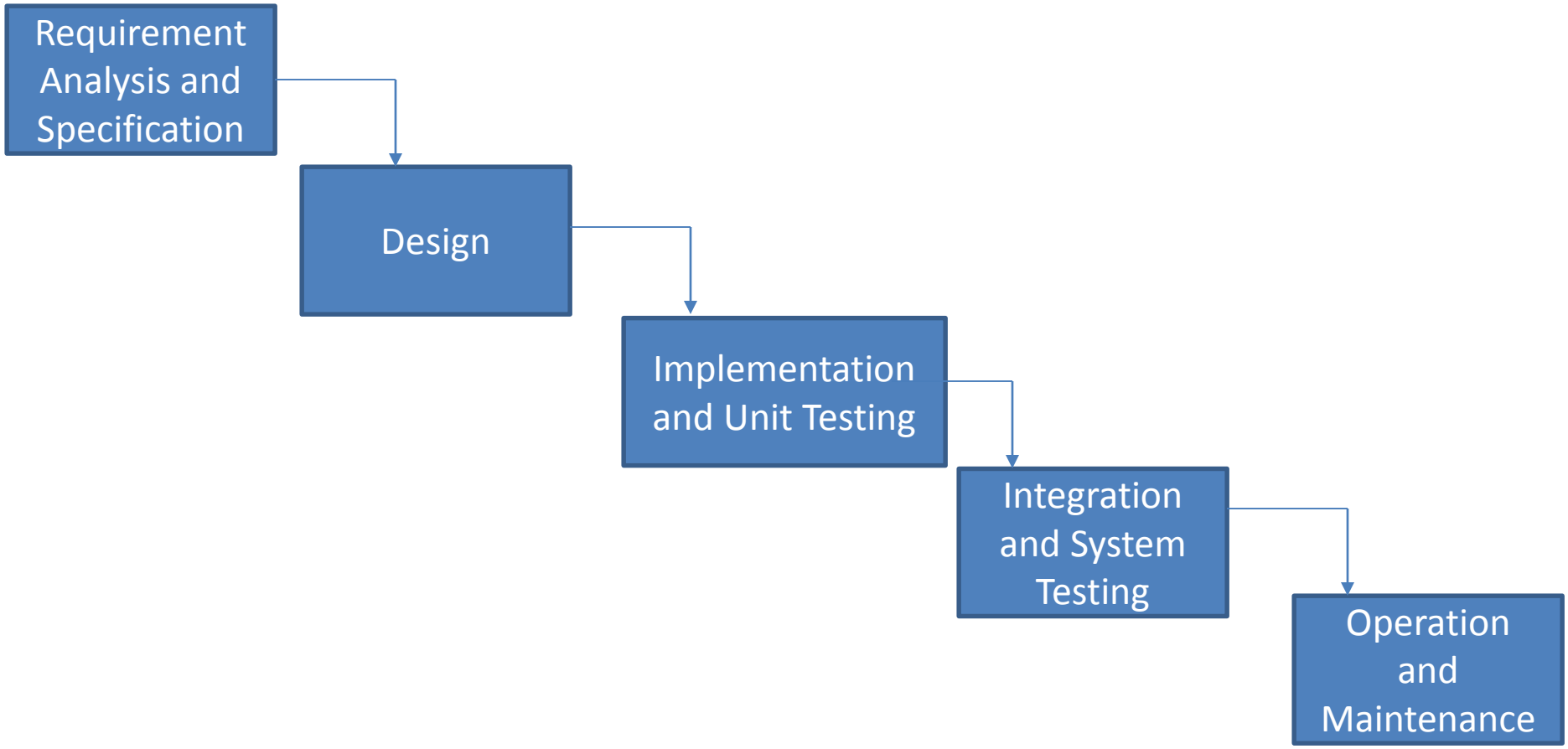
Name: Manka Sharma

Delivered on: 17th August 2020

Process Models

Software Development Life Cycle

- The Waterfall Model



Software Development Life Cycle

- The WaterFall Model
  - It has five phases i.e. Requirements Analysis, Design, Implementation and Unit testing, Integration and System testing, and operation and maintenance
  - It resembles waterfall and hence the name
- Requirement Analysis and Specification Phase:
  - To understand the exact requirements of the customer and document them properly.
  - This activity is executed with customer
  - The aim is to document all the functional and non functional requirements
  - The requirements describe “what” aspect of the system.

Software Development Life Cycle

- The resultant document is known as Software Requirement Specification document written in natural language.

- Design Phase:
  - The aim is to transform the requirements specifications into a structure suitable for implementation in programming language.
  - The work is documented in Software Design Document (SDD).
- Implementation and unit testing phase:
  - The aim is to convert the high level design to a working system.
  - The unit testing involves testing each module separately to localize errors.
  - Small modules are tested first in isolation

Software Development Life Cycle

- Integration and System Testing:
  - It is a very expensive activity and incurs one-third of the total cost.
  - The aim is to ensure that the whole system works accurately together.
- Operation and Maintenance
  - This is a broad activity and it includes error correction, enhancement of capabilities, deletion of obsolete capabilities and optimization
- Advantages of “The WaterFall Model”
  - Easy to Understand and explain to the users.
  - Works on “Define before Design” and “Design before Code” i.e. Structured approach.
  - Stages and activities are well defined.
  - Helps to plan and schedule the project.
  - Verification at each stage ensures early detection of errors/misunderstanding.
  - Each phase has specific deliverables.

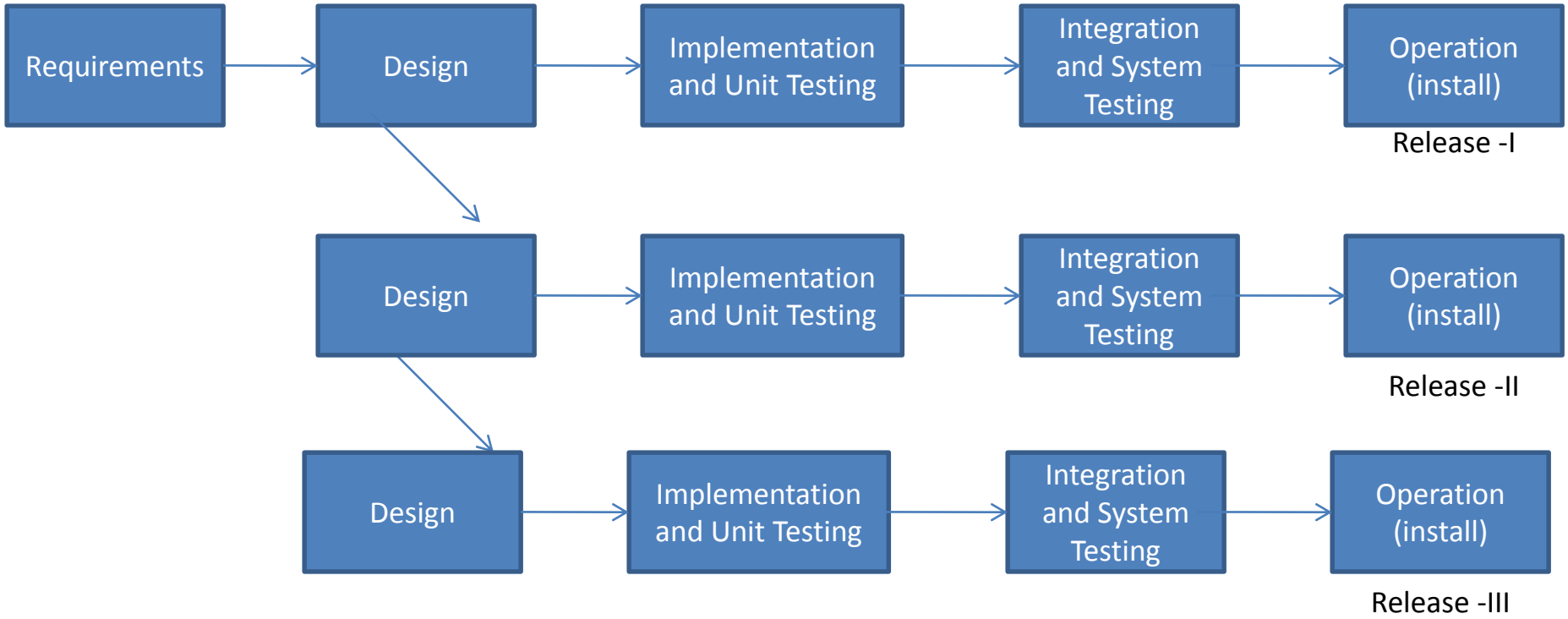
Software Development Life Cycle

- Disadvantages:
  - Complete requirements are expected to freeze early in the process
  - Working software is not available till late in the process
  - Accommodating changes is difficult
  - Very difficult to go back to any stage after it finished.
  - A little flexibility and adjusting scope is difficult and expensive.
  - Costly and required more time, in addition to the detailed plan.

Increment Process Models

- These are effective when requirements are defined precisely and there is no confusion about the functionality of the product
- After every cycle, a useable product is given to the customer
- Every new cycle adds a new functionality to the system
- It is more useful when a quick delivery of a limited functionality system needs to be developed
- These models include:
  - ✓ Iterative Enhancement Model
  - ✓ Rapid Application Development (RAD) Model

Iterative Enhancement Model



Iterative Enhancement Model

Iterative Enhancement Model

- Similar to waterfall model, but is based on several cycles.
- A useable product with added functionality is released at the end of each cycle
- During the requirements analysis phase, customers and developers consider as many requirements as possible and prepare SRS document.
- Requirements are then prioritized
- Developers then build the system in one or more cycles on defined priorities.

Iterative Enhancement Model

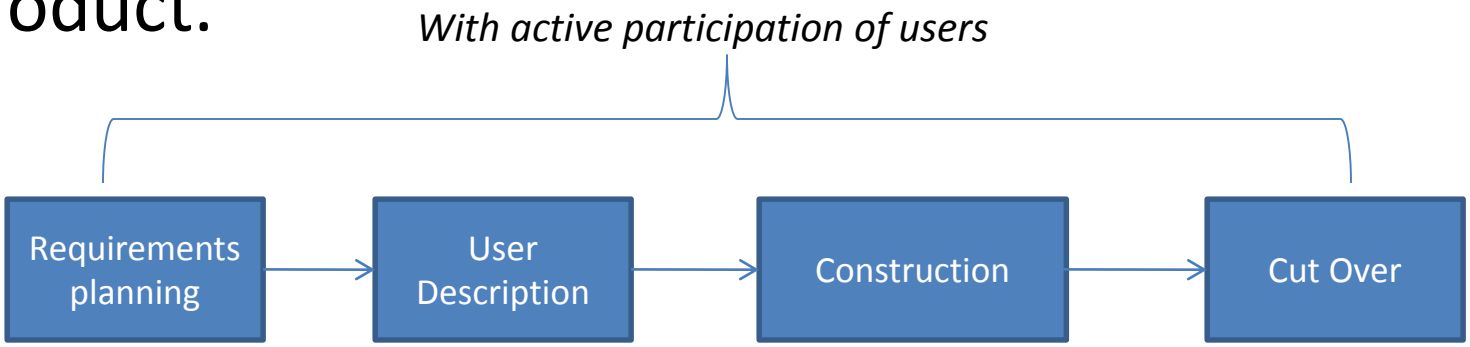
- The complete product is divided into releases.

Advantages:

- Produces business value early in the development lifecycle i.e. Some useful system may be given to the customer to start the work
- First release may be available within few weeks or months.
- Better use of scarce resources
- Can accommodate some change requests between increments.
- More focused on customer value than the linear approaches.
- Problems can be detected earlier.

Rapid Application Development (RAD)

- It is an increment process model, developed by IBM in 1980s.
- In this model, the user involvement is necessary right from requirements phase till the delivery of the product.



- It adjust the phases in such a way so as to get some part developed quickly and into the hands of the user.

Rapid Application Development (RAD)

- Requirements Planning: In this phase, requirements are gathered using group elicitation techniques. This is done for the active involvement of the users.
- User Description: Joint teams of developers and users are constituted to prepare, understand and review the requirements.
- Construction Phase: This phase combines the detailed design, coding and testing phases.
- Cut over Phase: It incorporates acceptance testing, installation and user training.

Rapid Application Development (RAD)

- As a result, the users better understand the system and suggest revision that brings system closer to the requirements.
- Most RAD prefer CASE tools, JAD technique, fourth generation programming language
- Advantages:
  - It is a combination of SDLC with tools and techniques to improve the speed and quality of the system developed
- Disadvantage:
  - At times it is difficult to match the user's expectations

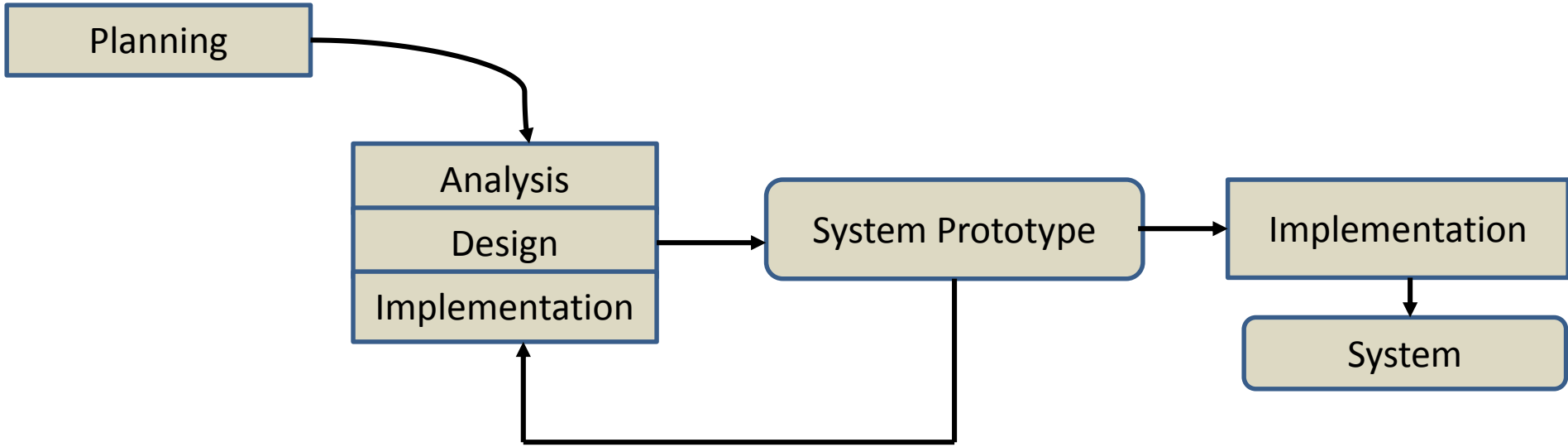
Prototyping Methodology

- Creating prototypes of software applications, for example, incomplete versions of the software program being developed
- It used to visualize some component of the software to limit the gap of misunderstanding the customer requirements by the development team.
- When the final prototype is developed, the requirement is considered to be frozen.
- Prototyping based methodology performs analysis, design and implementation phases concurrently

Prototyping Methodology

- These phases are performed repeatedly in a cycle until the system is completed.
- In this, the work immediately begins to develop a system prototype, i.e. “quick-and-dirty” program that provides minimal amount of features
- The first prototype created is shown to the users and their reaction and comments are gathered.
- Based on the feedback collected, the system is reanalyzed, redesigned and re-implemented to produce second prototype.

Prototyping Methodology



Prototyping Methodology

- This process is repeated until the prototype provides enough functionality to be installed and used in the system.

Advantage :

- It quickly provides the system for the users to interact with, even if it is not initially ready for widespread organizational use.
- It reassures the users that the project team is working on the system and the approach helps to more quickly refine real requirements.

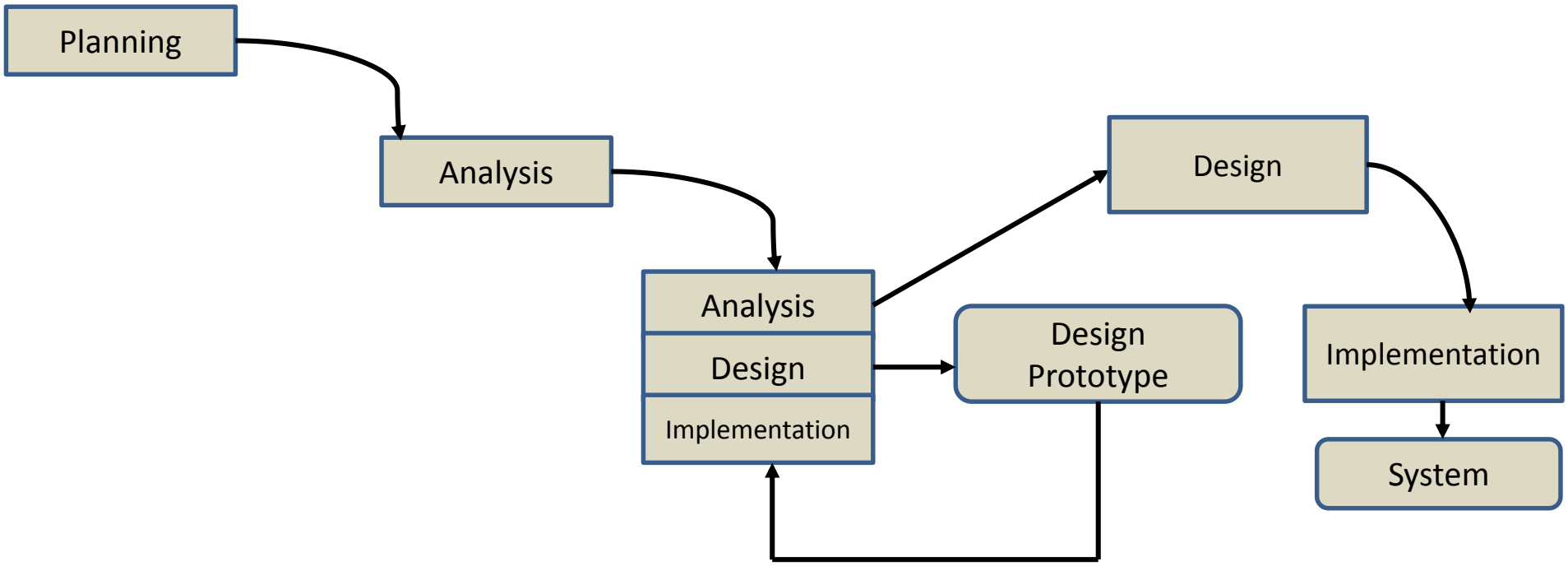
Prototyping Methodology

Disadvantage:

- The fast paced system releases may prevent conduct careful, methodical analysis.
- The prototype may undergo several changes and the initial design decisions prove to be not the good ones.

Throwaway Prototyping Methodology

- This methodology has relatively thorough analysis phase.



Throwaway Prototyping Methodology

- It builds a design prototype. A design prototype is not a working system, it is a product that represents a part of the system that needs additional refinement.
- A design prototype contains enough detail to enable users to understand the issues under consideration.
- Several design prototypes are formed.
- This reduces the risk associated with the system as important issues are discussed before the real system is built.
- Prototypes that are eventually discarded rather than becoming a part of the finally delivered software

Throwaway Prototyping Methodology

Advantages:

- It balances the benefits of well thought out analysis and design phases with the advantages of using prototypes to refine key issues before a system is built.
- It produces more stable and reliable systems.
- Reduced time and costs, but this can be a disadvantage if the developer loses time in developing the prototypes.
- Improved and increased user involvement.

Disadvantage:

- It may take longer to deliver the final system as compared with the prototyping based methodology.

References

- Bob Hughes and Mike Cotterell, “Software Project Management ”, Tata McGraw Hill, 4th edition, 2006 .
- Software Project Management, Tutorialspoint. https://www.tutorialspoint.com/software_engineering/software_project_management.htm (accessed on 18th July 2020).