

Information Systems (තොරතුරු පද්ධති)

Lesson 02

(Systems Development Models
and Methods)



Objectives – Learning Outcomes

- The objectives of this section are to introduce system development models and methods.

When you have followed this section you will:

- ✓ Understand what system development models, its stages, merits/demerits etc.
 - Waterfall
 - Spiral
 - Unified development
 - Rapid Applications Development



Outline

- What is a system development model?
- Current system development Models
- Merits and Demerits

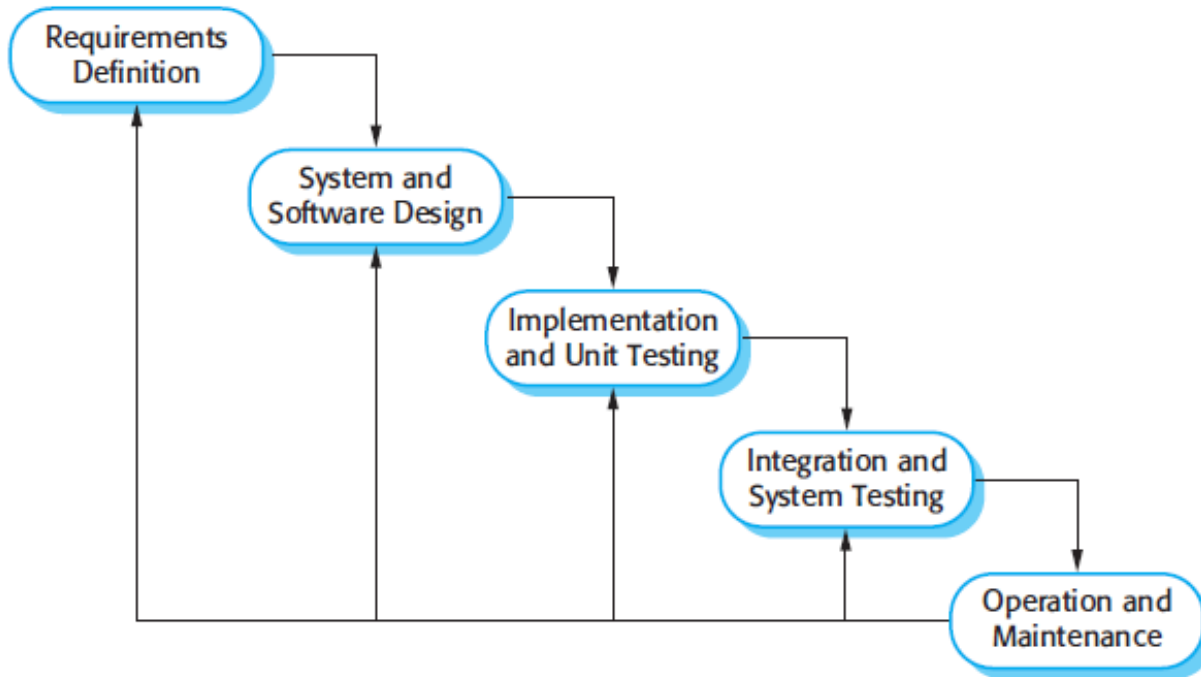


What is a system development model?

- A system development model is a simplified representation of a system development process
- Each process model represents a process from a particular perspective, and thus provides only partial information about that process
- Examples – Waterfall model, Spiral Model etc.



❑ Waterfall model



The **waterfall model** is a sequential design process, used in software/system development processes, in which progress is seen as flowing steadily downwards (like a **waterfall**) through the phases of it

Waterfall model Cont.

1. Requirements analysis and definition

The system's services, constraints, and goals are established by consultation with system users.

They are then defined in detail and serve as a system specification.

2. System and software design

The systems design process allocates the requirements to either hardware or software systems by establishing an overall system architecture.

Software design involves identifying and describing the fundamental software system abstractions and their relationships.



Waterfall model Cont.

3. Implementation and unit testing

During this stage, the software design is realized as a set of programs or program units.

Unit testing involves verifying that each unit meets its specification.

4. Integration and system testing

The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met.

After testing, the software system is delivered to the customer.



Waterfall model Cont.

5. Operation and maintenance

Normally, this is the longest life cycle phase.

The system is installed and put into practical use.

Maintenance involves correcting errors which were not discovered in earlier stages of the life cycle, improving the implementation of system units and enhancing the system's services as new requirements are discovered.

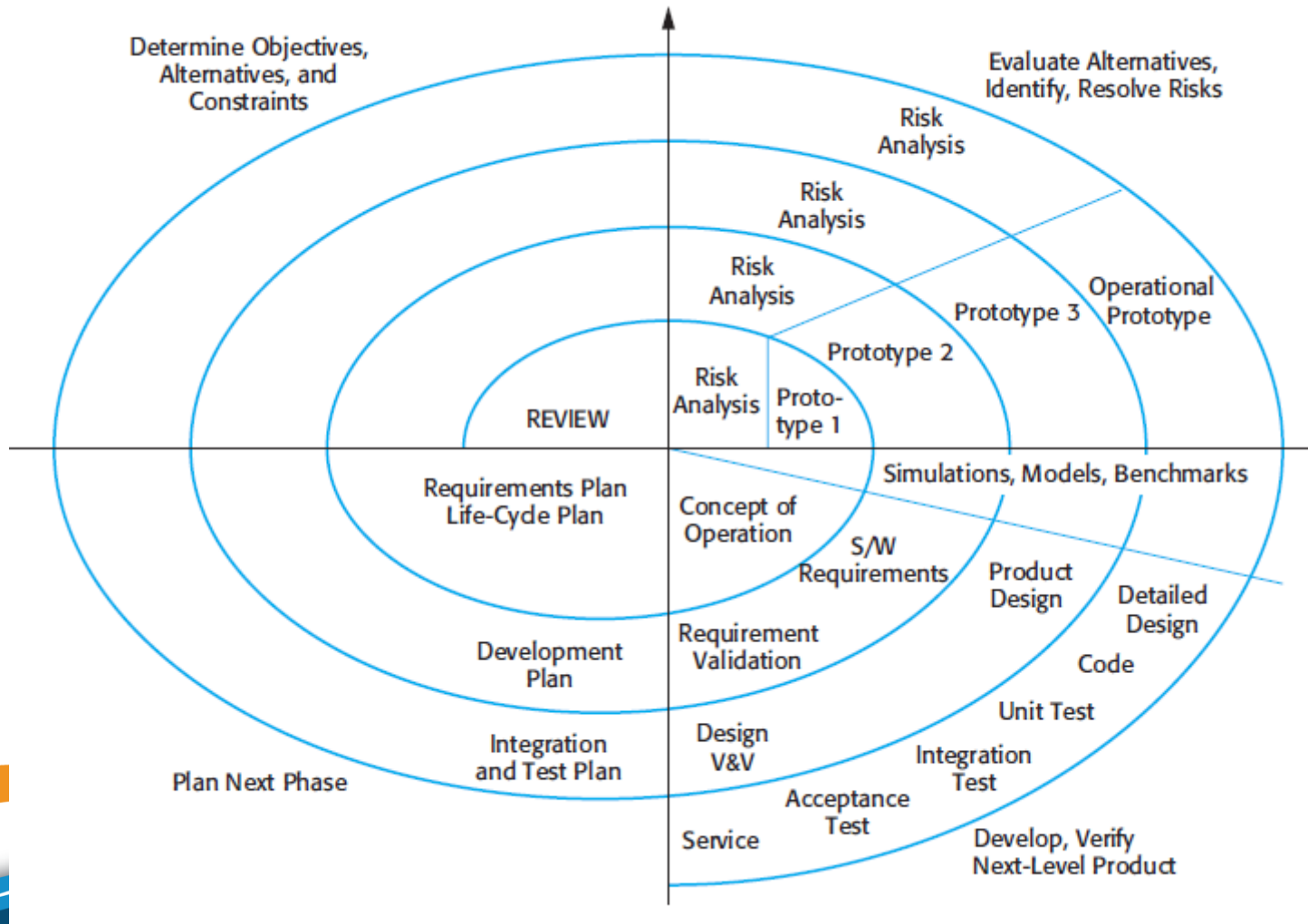


❑ Spiral model

- The software process is represented as a spiral, rather than a sequence of activities with some backtracking from one activity to another
- Each loop in the spiral represents a phase of the software process



Spiral model



Spiral model Cont.

1. Objective setting - Specific objectives for that phase of the project are defined. Constraints on the process and the product are identified and a detailed management plan is drawn up. Project risks are identified. Alternative strategies, depending on these risks, may be planned.

2. Risk assessment and reduction - For each of the identified project risks, a detailed analysis is carried out. Steps are taken to reduce the risk. For example, if there is a risk that the requirements are inappropriate, a prototype system may be developed.



Spiral model Cont.

3. *Development and validation* - After risk evaluation, a development model for the system is chosen. For example, throwaway prototyping may be the best development approach if user interface risks are dominant. If safety risks are the main consideration, development based on formal transformations may be the most appropriate process, and so on. If the main identified risk is sub-system integration, the waterfall model may be the best development model to use.

4. *Planning* - The project is reviewed and a decision made whether to continue with a further loop of the spiral. If it is decided to continue, plans are drawn up for the next phase of the project.

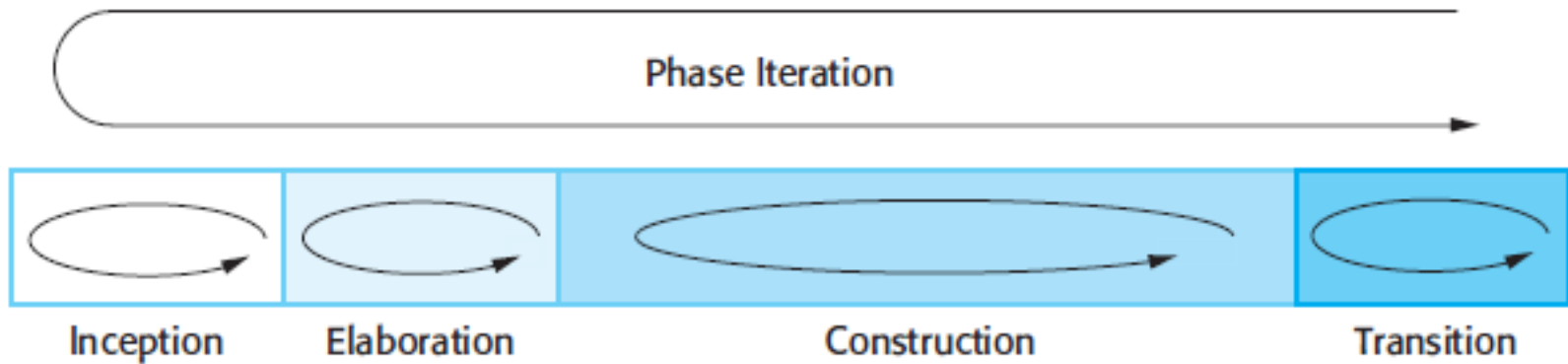


❑ Unified Development model

- The Unified Process is a traditional “cathedral” style of incremental design driven by constructing views of a system architecture. It has the following key features:
 - It is component based, commonly being used to coordinate object oriented programming projects.
 - It uses UML - a diagrammatic notation for object oriented design - for all blueprints.
 - The design process is anchored, and driven by, use-cases which help keep sight of the anticipated behaviors of the system.
 - It is architecture centric. Design is iterative and incremental - via a prescribed sequence of design phases within a cyclic process.



Unified Development model



Unified Development model- phases

Inception : Produces a commitment to go ahead. By the end of this phase a business case should have been made; feasibility of the project assessed; and the scope of the design should be known.

Elaboration : Takes us to a working specification of the system. By the end of this phase a basic architecture should have been produced; a plan of construction agreed; all significant risks identified; and those risks considered to be major should have been addressed.

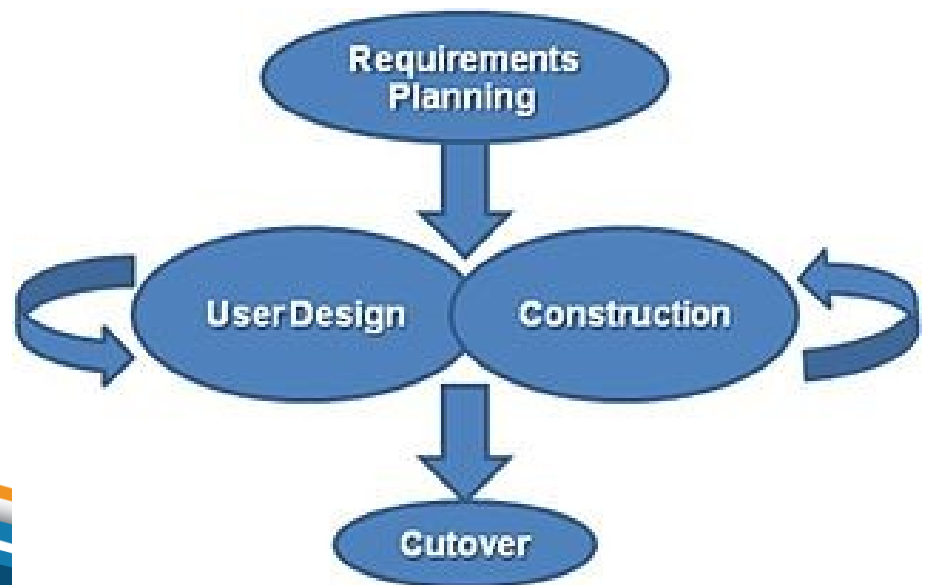
Construction : Produces beta-release system. By the end of this phase a working system should be available, sufficient for preliminary testing under realistic conditions.

Transition : Introduces the system to its intended users.



❑ Rapid Application Development model

- An approach to software/system development aimed at rapid delivery of the software.
- It often involves the use of database programming and development support tools such as screen and report generators.



Rapid Application Development model Cont.

- **Requirements planning phase** – Combines elements of the system planning and systems analysis phases of the Systems Development Life Cycle (SDLC). Users, managers, and IT staff members discuss and agree on business needs, project scope, constraints, and system requirements. It ends when the team agrees on the key issues and obtains management authorization to continue.
- **User design phase** – During this phase, users interact with systems analysts and develop models and prototypes that represent all system processes, inputs, and outputs. The RAD groups or subgroups typically use a combination of Joint Application Development (JAD) techniques and CASE tools to translate user needs into working models. *User Design* is a continuous interactive process that allows users to understand, modify, and eventually approve a working model of the system that meets their needs.



Rapid Application Development model Cont.

- **Construction phase** – Focuses on program and application development task similar to the SDLC. In RAD, however, users continue to participate and can still suggest changes or improvements as actual screens or reports are developed. Its tasks are programming and application development, coding, unit-integration and system testing.
- **Cutover phase** – Resembles the final tasks in the SDLC implementation phase, including data conversion, testing, changeover to the new system, and user training. Compared with traditional methods, the entire process is compressed. As a result, the new system is built, delivered, and placed in operation much sooner



Merits & Demerits - Waterfall Model

Merits	Demerits
<ul style="list-style-type: none">• Straightforward, simple and easy to understand & use	<ul style="list-style-type: none">• All requirement need to be known upfront
<ul style="list-style-type: none">• Appropriate for short-term projects	<ul style="list-style-type: none">• Not much rooms for revisions and inflexible
<ul style="list-style-type: none">• Well-suited for projects with clear, stable and direct plans	<ul style="list-style-type: none">• Cannot move onto the next step until the prior step is completed
<ul style="list-style-type: none">• Provide structure for inexperienced staff	<ul style="list-style-type: none">• Cannot solve problems until the maintenance phase
<ul style="list-style-type: none">• Good for executive and management control	<ul style="list-style-type: none">• Little opportunity to preview the system until integration at the very end
<ul style="list-style-type: none">• Each stage has specific deliverables	



Merits & Demerits - Spiral Model

Merits	Demerits
<ul style="list-style-type: none">• Changing requirements can be accommodated	<ul style="list-style-type: none">• Management is more complex
<ul style="list-style-type: none">• Allows for extensive use of prototypes	<ul style="list-style-type: none">• End of project may not be known early
<ul style="list-style-type: none">• Requirements can be captured more accurately	<ul style="list-style-type: none">• Not suitable for small or low risk projects and could be expensive for small projects
<ul style="list-style-type: none">• Users see the system early	<ul style="list-style-type: none">• Process is complex
<ul style="list-style-type: none">• Development can be divided into smaller parts and more risks parts can be developed earlier which helps better risk management	<ul style="list-style-type: none">• Spiral may go indefinitely
	<ul style="list-style-type: none">• Large number of intermediate stages requires excessive documentation

Merits & Demerits – Unified Dev. Model

Merits	Demerits
<ul style="list-style-type: none">• Unified Process supports multiple architectural models and views	<ul style="list-style-type: none">• Management is more complex
<ul style="list-style-type: none">• One of the most important deliverables of the process is the executable architecture baseline which is created during the Elaboration phase	<ul style="list-style-type: none">• Process is complex
<ul style="list-style-type: none">• Greatest risks are addressed first	



Merits & Demerits - RAD Model

Merits	Demerits
Measurable Progress	Requires Modular Systems
Quickly Generate Productive Code	Difficulty Within Large-Scale Projects
Compartmentalization of System Components	Demands Frequent User Interfacing
Rapid, Constant User Feedback	Depends Upon Skilled Developers
Early Systems Integration	
Simple Adaptability	



Summery

In this section you have given an idea about the systems development models and methods.

Also this section covered the details about below models with merits and demerits – Waterfall, Spiral, Unified development & Rapid Applications Development Models



References

- SOFTWARE ENGINEERING, Ninth Edition, Ian Sommerville

Next Lesson >>>

**Lesson 03 : Structured System Analysis and
Design Methodology**

