Software Engineering

Er. Prahlad Panthi

Software Validation

- To show that a system both conforms to its specification and that it meets the expectations of the system customer.
- Stages in testing process:
  - **Development testing**
    - Each component is tested independently
    - Testing is carried out by developers of that system or component
  - **System testing**
    - Testing whole system after integrating all the components
  - **Acceptance testing**
    - System is tested with real data rather than simulated test data
    - Also known as *alpha test*

Note: "*Beta Test"* term is used when a system is to be marketed as a software product

Software Evolution (Software Maintenance)

- Maintaining or evolving a software system after it is being delivered to the customer.
- Costs of maintenance are often several times the initial development costs.
- Maintenance processes are sometimes considered to be less challenging than the original software development

Coping with change

- Change adds to the costs of software development.
- It may be necessary to redesign the system to deliver the new requirements, change any programs that have been developed and re-test the system.
- Two related approaches that may be used to reduce the costs of rework are:
  - Change avoidance
    - Anticipate possible changes before significant rework is required
    - Prototype system may be developed to show some key features of the system to customers. They can experiment with the prototype and refine their requirements before committing to high software production costs.

  - Change tolerance
    - Process is designed so that changes can be accommodated at relatively low costs
    - Normally involves some form of incremental development

- Two ways of coping with change:
  - System Prototyping (supports change avoidance)
  - Incremental delivery (supports both change avoidance and change tolerance)

Prototyping

- A prototype is the initial version of software system that is used to demonstrate concepts, try out design options and find out more about the problem and its possible solutions.
- In Requirement Engineering a prototype can help with elicitation and validation of system requirements.
- In system design process, prototype can be used to explore particular software solution and support User Interface (UI) design.
- Prototypes don’t have to be executable programs, they can be mock-ups of system interface.

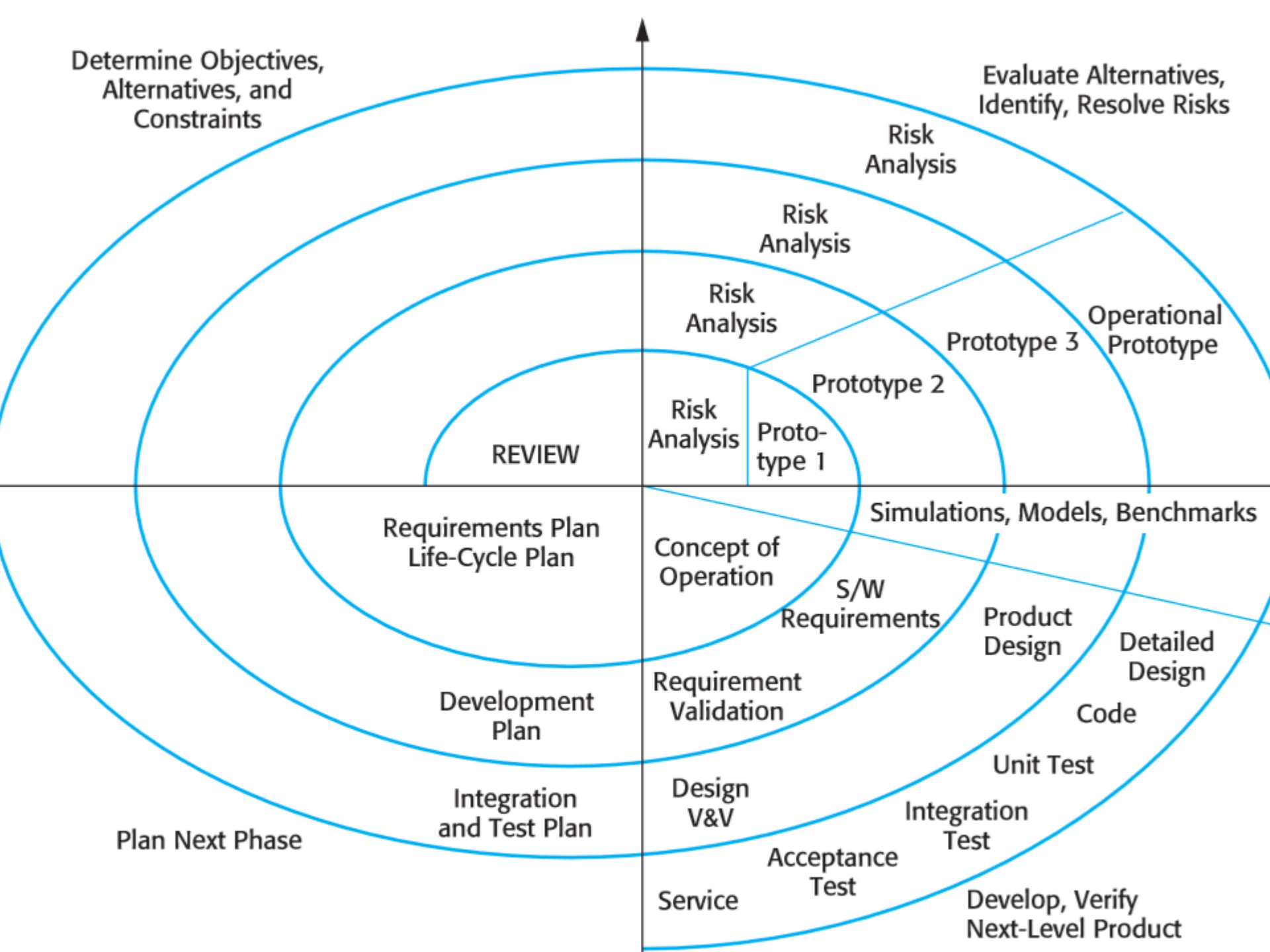
Incremental Delivery

- As new increments are completed, they are integrated with existing increments so that system functionality improves with each delivered increment.
- Advantages:
  - Customer can use early increments as prototypes and gain experience that informs their requirements for later system increments. (These are real system unlike in prototyping method)
  - Customer do not have to wait until the entire system is delivered before they can gain value from it.
  - Highest-priority services are delivered first, so customers are less likely to encounter software failures in most important parts of system.

- Problems:
  - System requires a set of basic facilities that are used by different parts of the system.
  - It is difficult when replacement systems are being used by customers as they demand a complete system before replacing old one.

Spiral Model

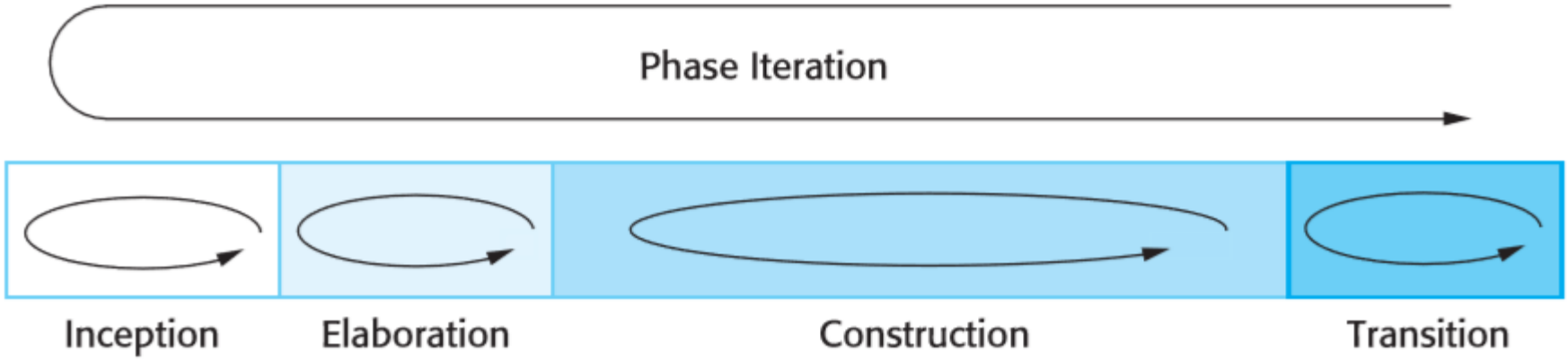
- Risk-driven software process framework, where each loop represents a phase of software process.
- Assumes that changes are a result of project risks and includes explicit risk management activities to reduce these risks.
- Each loop in the spiral is split into four sectors:
  - Objective Setting
    - Objectives for that phase of the project are defined.
    - Constraints on the process and the product are identified and detailed management plan is drawn.
    - Project risks are identified with alternative strategies.
  - Risk assessment and reduction
  - Development and validation (development model for system is chosen)
  - Planning (project is reviewed and a decision is made whether to continue with a further loop of the spiral)


Determine Objectives, Alternatives, and Constraints
Evaluate Alternatives, Identify, Resolve Risks
Risk Analysis
Risk Analysis
Risk Analysis
Risk Analysis
Operational Prototype
Prototype 3
Prototype 2
Proto-type 1
REVIEW
Simulations, Models, Benchmarks
Requirements Plan
Life-Cycle Plan
Concept of Operation
S/W Requirements
Product Design
Detailed Design
Code
Unit Test
Integration Test
Acceptance Test
Development Plan
Requirement Validation
Design V&V
Integration and Test Plan
Plan Next Phase
Service
Develop, Verify Next-Level Product

Rational Unified Process (RUP)

- Designed in conjunction with Unified Modeling Language (UML).
- Normally described from three perspectives:
  - Dynamic perspective: shows phases of model over time.
  - Static perspective: shows process activities that are enacted.
  - Practice perspective: suggests good practices to be used during the process.
- Has 4 phases which are closely related to business rather than technical concerns.
- Has 6 core workflows and 3 supporting work-flows.
- Not suitable for all types of development. Example: embedded software development.

Phases


Phase Iteration
Inception
Elaboration
Construction
Transition

- Inception
  - Establish a business case for the system.
  - Identify all external entities (people and systems) that will interact with the system and define these interactions.
  - Use this information to access the contribution that the system makes to the business.
  - If this contribution is minor, then the project may be cancelled after this phase.
- Elaboration
  - Main goal is to develop an understanding of the problem domain, establish an architectural framework for the system, develop the project plan and identify key project risks.
- Construction
  - System design, programming and testing
- Transition
  - Moving the system from developer community to user community and making it work in real environment.

- Iteration within RUP is supported in two ways.
  - Each phase may be enacted in an iterative way with the results developed incrementally.
  - Whole set of phases may also be enacted incrementally.
- Static view of RUP focuses on the activities that take place during the development process. These are called workflows in RUP description.
- There are six core workflows and three supporting workflows.
- Workflows are static and are technical activities that are not associated with a single phase but may be used throughout the development to achieve the goals of each phase.
- It is not suitable for all types of development. Example: embedded system software development.

Workflows

Workflow	Description
Business modelling	The business processes are modelled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models, and sequence models.
Implementation	The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process.
Testing	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users, and installed in their workplace.
Configuration and change management	This supporting workflow manages changes to the system (see Chapter 25).
Project management	This supporting workflow manages the system development (see Chapters 22 and 23).
Environment	This workflow is concerned with making appropriate software tools available to the software development team.

Six fundamental best practices:

- Develop software iteratively
- Manage requirements
- Use component based architectures
- Visually model software to present static and dynamic views
- Verify software quality
- Control changes to software

To Be Continued…