AGILE-PARADIGM SHIFT IN SDLC

Session - 4

Iterative Process Models

Objectives

- Explain prototype model
- List the advantages and disadvantages of prototyping
- Explain prototype model suitability in project development
- Explain incremental process model and its approach
- Explain iterative process benefits and drawbacks
- Explain spiral model
- Describe the phases in the spiral model
- List the applications of spiral model
- Explain the advantages and disadvantages of the spiral model

Introduction

- Some of the drawbacks of Waterfall model are:
  - The software development team has to gather all the requirements in the beginning of the project.
  - The risk of failure is quite enormous in such a case, as defects results in costly over-runs and in some cases even project cancellation.

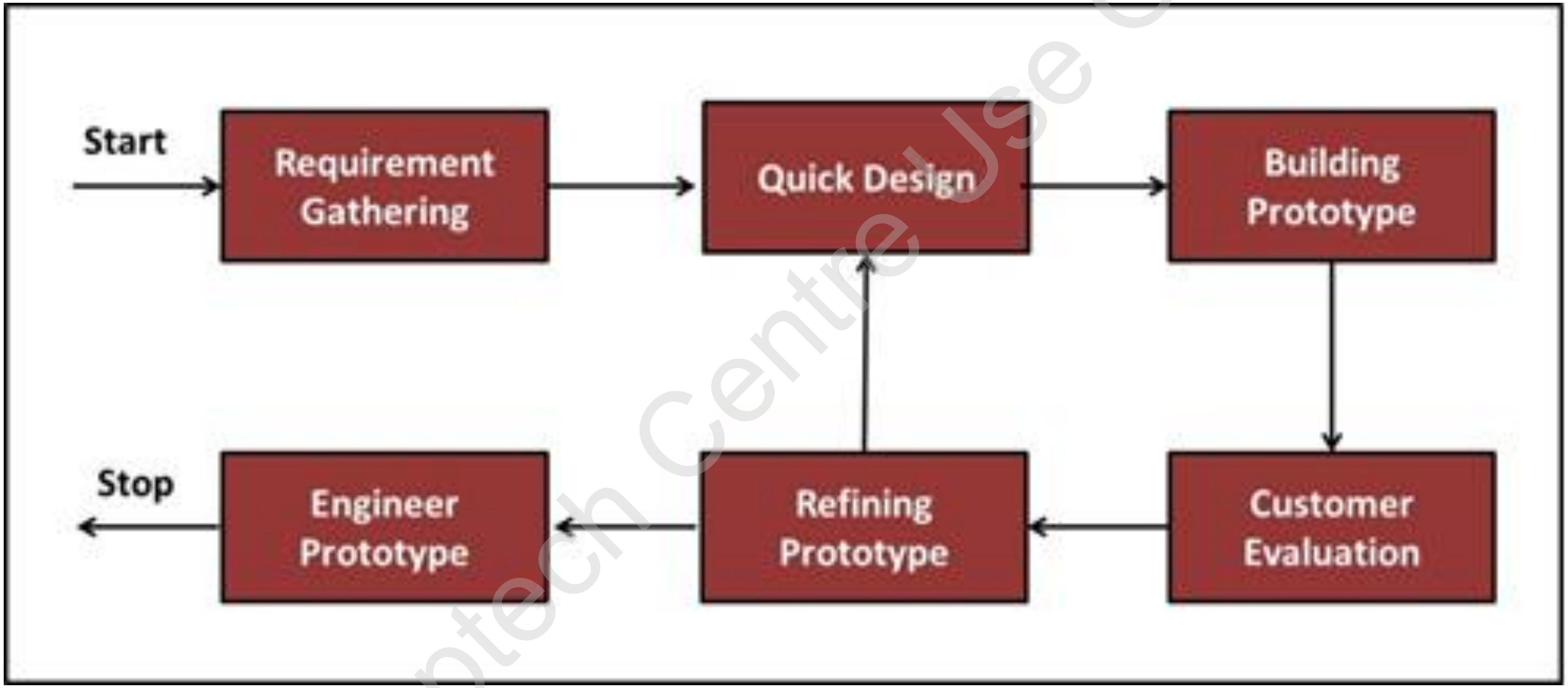
The drawbacks of the Waterfall model resulted in an **iterative process models**.

Prototype Model [1-2]

- It was developed to overcome the limitations of the Waterfall model.
- A prototype:
  - Is created based on the existing requirements.
  - Is simulation of the real thing and displays the functionality of the product under development.
  - Helps to get valuable feedback from the customer.
  - Enables software designers and developers learn about what is expected from the desired product.

Prototype Model [2-2]

- Figure depicts the various phases of prototype model.



Steps to Design a Software Prototype

- The approach taken to design a software prototype is as follows:

Identification of Basic Requirements

- Requirements in terms of user interface is identified.
- Extensive details such as internal design, performance, and security are not considered.

Development of the Initial Prototype

- Very basic requirements are illustrated and user interfaces are designed.

Review of the Prototype

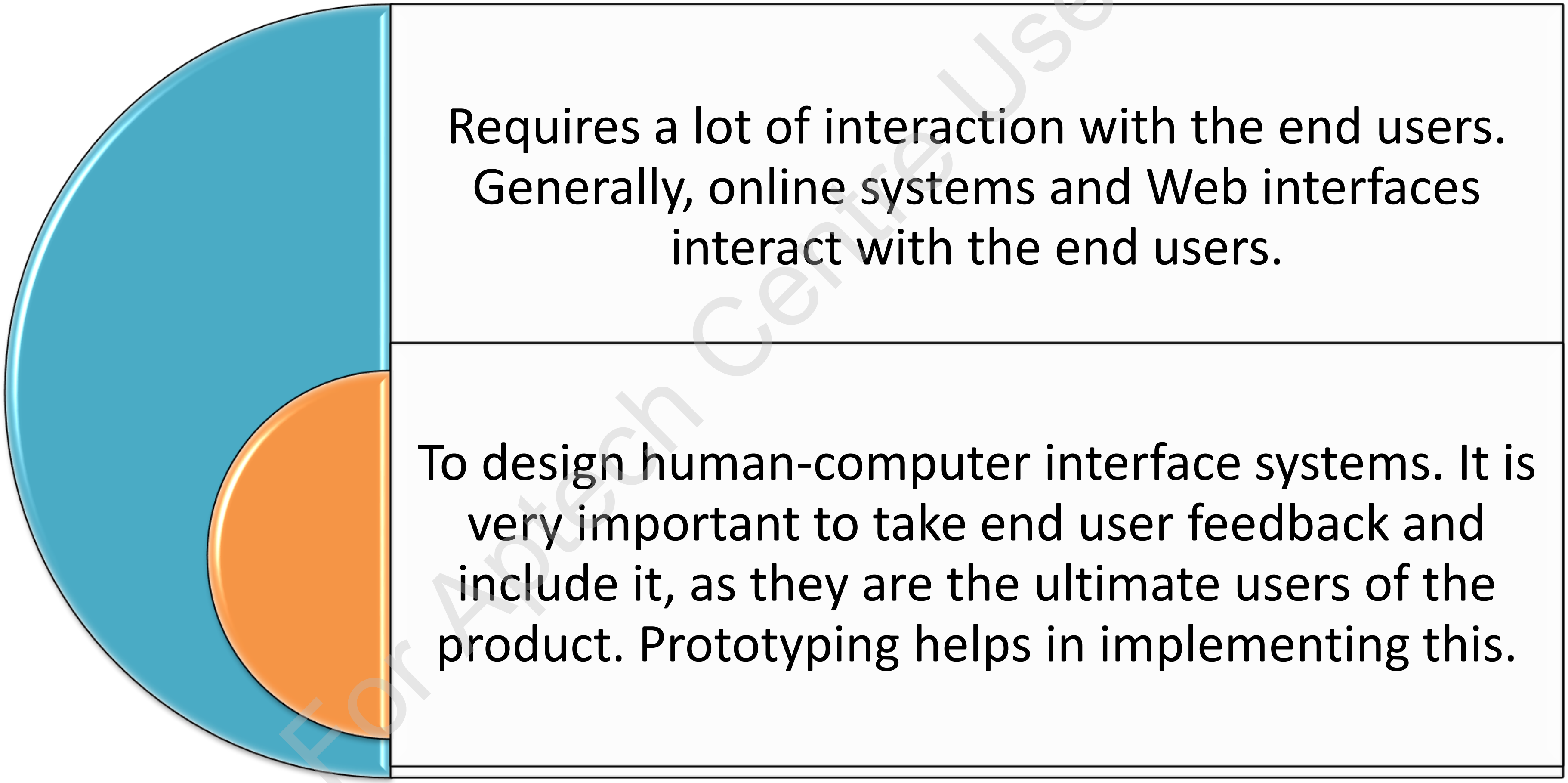
- Developed prototype is presented to the customer for review.
- Feedback is collected and documented.

Revision and Enhancement of the Prototype

- Feedback and the review comments from the customer are evaluated.
- Accepted changes are incorporated into the new prototype developed and the cycle repeats until customer expectations are met.

Applications of Prototype Model

- The prototype model is used for projects with the following criteria:



Requires a lot of interaction with the end users. Generally, online systems and Web interfaces interact with the end users.

To design human-computer interface systems. It is very important to take end user feedback and include it, as they are the ultimate users of the product. Prototyping helps in implementing this.

Deciding Whether to Prototype or Not

- Table lists the situations and corresponding prototype model considerations.

Situation	Prototype
Users are uncomfortable with hypothetical models.	Gives users a prototype to interact with the model, before actual product.
Project has a long development time.	Gives users and developers something to work in the early phase of the development cycle.
Requirements are uncertain.	Provides opportunities to rework on requirements.
A complete new system with no earlier history.	Provides chances to do multiple trials, detect and correct errors.
Product development needs the help of tools such as simulation, experimentation, or progressive assessment.	Allows for simulation, experimentation, and incremental evaluation.
A critical system is needed immediately.	Prototyping tools are designed for quick implementation. Can begin requirements gathering soon.
Users are available.	Allows for high user participation.

Advantages/Disadvantages of Prototype

- Some of the advantages of prototype model are as follows:
  - There is active involvement of users in the project.
  - The working prototype model gives a better understanding of the desired product.
  - The problems that can arise during the development are detected earlier.
  - The users feedback helps in providing better solutions.
  - The complex modules are identified and a quick implementation can be provided for them.
- Some of the disadvantages of prototype model are as follows:
  - Excessive change requests are one of the major concerns.
  - Users tend to request changes repeatedly, just because of the nature of the iterative process.

Evolutionary Process Model [1-2]

- Factors such as:
  - **Business** and **product** requirements often change as development proceeds, making a straight line path to an end product unrealistic.
  - **Tight market deadlines** make completion of a comprehensive software product impossible.
- The evolutionary nature of software is not considered in this classic software engineering paradigms.

The drawbacks of the linear sequential models resulted in **Evolutionary Models** that are **iterative**.

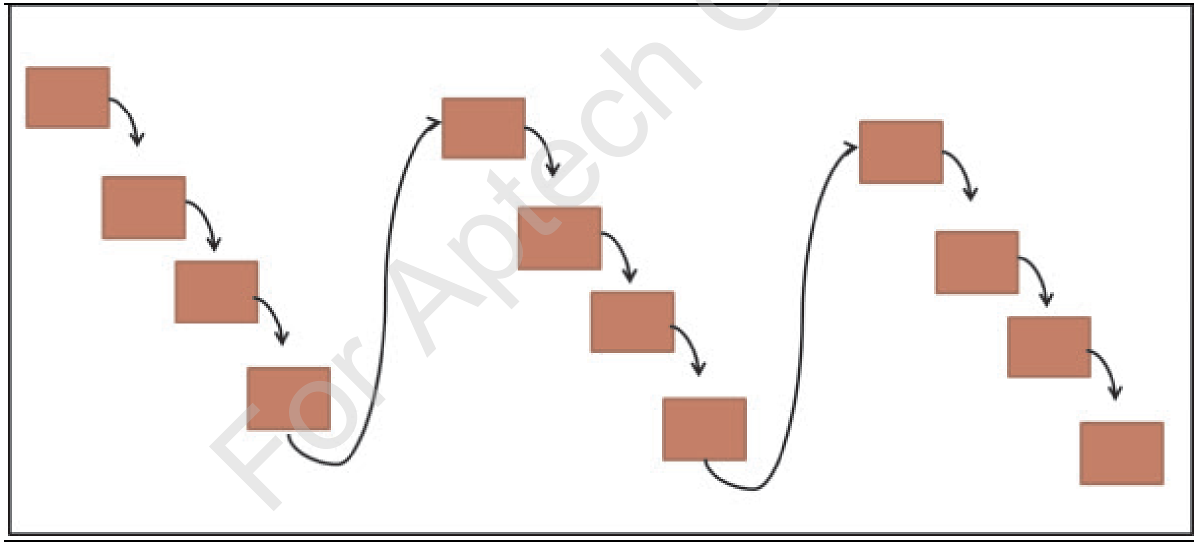
Evolutionary Process Model [2-2]

- Evolutionary Models:
  - Are based on iterative approach.
  - Are characterized in a manner that enables software engineers to develop increasingly more complete versions of the software.
  - Are time-based.

IMPLEMENT
DEVELOPMENT
ANALYZE
DESIGN
canstock
© Can Stock Photo - csp3430487

Iterative Approach [1-2]

- Development begins by determining and implementing a part of the software project, which is then reviewed to identify further requirements.
- The process is then repeated and a new version of the software is created for each cycle of the model.
- Every iteration acts like a mini Waterfall model and the feedback from one phase is fed as an input to the next phase.
- At the end of each iteration, working software is delivered which is fed to the next iteration.
- Figure shows the iterative model in the software development.



Iterative Approach [2-2]

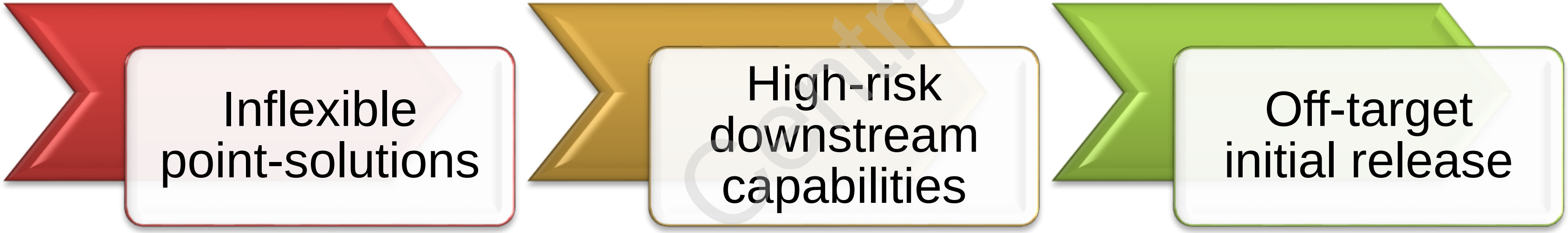
- The main features of iterative development are:
  - Time-boxed phases
  - Test early, test often
  - Deliver early, deliver often
  - Production quality

Benefits of Iterative Approach

- An iterative approach is generally superior to a linear or Waterfall approach for the following reasons:
  - Tolerates changing requirements
  - Elements are integrated progressively
  - Risks are mitigated earlier
  - Allows the organization to learn and improve
  - Facilitates reuse
  - Results in a more robust product
  - Tolerates tactical changes in the product
  - Process can be improved and refined along the way
  - Accommodates changes
  - Allows technological changes on the way
  - Increases reuse
  - Easier to take advantage of Commercial-Off-The-Shelf-Products
  - Learning along the way
  - Results in a more thoroughly tested product

Problems with Iterative Approach

- The ’**initial release**’ milestone frequently has the following problems:



Inflexible point-solutions

High-risk downstream capabilities

Off-target initial release

Spiral Model [1-3]

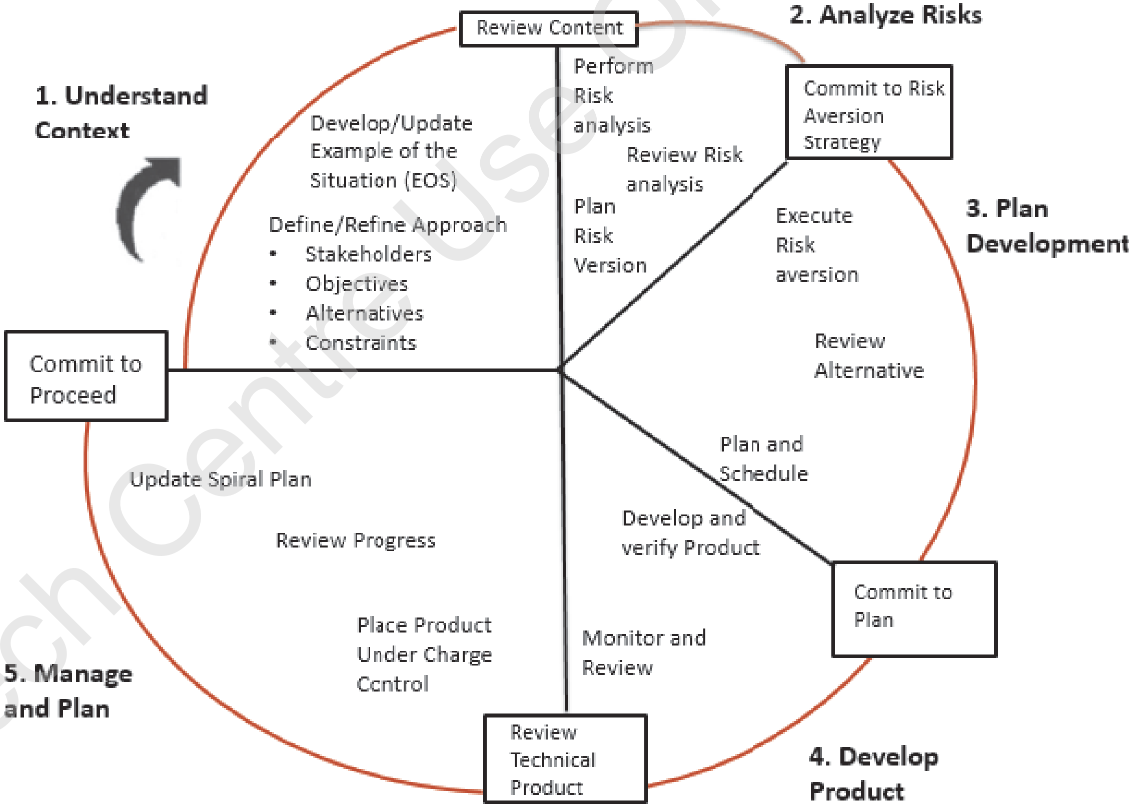
- Boehm proposed the Spiral model in 1988 as an evolutionary software process model.

Spiral Model

- Couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model.
- Views software development as a spiral process with the software being developed in a series of incremental releases.
- Attempts to combine the strengths of the various models.
- Provides the potential for rapid development of incremental versions of the software.
- Takes into account results of risk assessment activities whose outcome determines taking up of the next phase in the development activity.

Spiral Model [2-3]

- Figure shows the Spiral model.
    - The inner cycles represent the early phases of requirement analysis along with prototyping to refine the requirement definitions.
    - The outer spirals are progressively representative of the classic software life cycle.
- The spiral model is divided into a number of framework activities, also called **task regions**.



Typically, there are between three to six **task regions**

Spiral Model [3-3]

- The six task regions in the spiral model are:

Understanding the context	These tasks are required to establish effective communication between developer and customer.
Planning tasks	They define resources, timelines, and other project related information.
Risk analysis tasks	These are required to assess both technical and management risks.
Engineering tasks	These are required to build one or more representations of the application.
Construction and release	These tasks construct, test, install, and provide user support.
Customer evaluation tasks	These are required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

Advantages/Disadvantages of Spiral Model

- Advantages:
  - Better understanding of risks at evolutionary level.
  - Realistic approach to the development of large-scale systems and software.
  - Adopts systematic approach of the classical life cycle, but incorporates it into an iterative framework that more realistically reflects the real word.
- Disadvantages:
  - It is not a solution to all the problems faced by software developers.
  - It demands considerable risk assessment expertise, and relies on this expertise for success.
  - If major risks are not uncovered and managed, problems will undoubtedly occur.

Applications of Spiral Model

- Spiral model is very widely used in the software industry.
- Spiral model is chosen for projects possessing the following criteria:
  - Medium to high-risk projects.
  - Long-term project commitment due to changes in economic priorities as requirements change with time.
  - Customer is not sure of their requirements.
  - Complex requirements, which further require clarification.
  - New product line, which needs to be released in phases to get enough customer feedback.
  - Major changes are expected in the product during the development life cycle.

Summary [1-2]

- The drawbacks of the Waterfall model resulted in an iterative process models which ensured faster results, with less information in advance, and much more flexible.
- Prototypes are similar to models or simulation of the real thing and may display the functionality of the product under development.
- In prototype, excessive change requests are one of the major concerns. Users tend to request changes repeatedly, just because of the nature of the iterative process.
- Evolutionary models are iterative. They enables software engineers to develop increasingly more complete versions of the software.

Summary [2-2]

- The key to successful use of an iterative software development life cycle is correct validation of requirements, verification, and testing of each version of the software against the requirements in each cycle of the model.
- The spiral model is as an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model.
- The spiral model is not a solution to all the problems faced by software developers. It may be difficult to convince customers (particularly in contract situations) that the evolutionary approach is controllable.