

Software Development Methodologies - Part II

LECTURE 05

MR. ALFRED EDWIN WORTHINGTON

Learning Outcomes

End of this lecture you will be able to learn,

- LO1: Ability to understand how to apply SDLC for Rapid Application Development.
- LO2: To explain how an iterative, incremental development process leads to faster delivery of more useful software.
- LO3: To explain the role of prototyping in the software process.

Rapid Application Development (RAD)

- Rapidly **changing business environments** have to respond **to new opportunities and competition.**
- RAD targets at **developing software in a short span of time.**
- Emphasizes **rapid prototyping** and **iterative delivery.**
- Can make **multiple iterations and updates** to a software rapidly without needing to start from scratch each time.

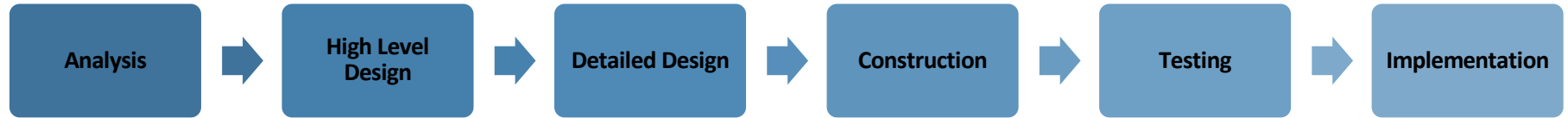
Why not with traditional Waterfall approach?

- Impossible to arrive at a stable, consistent set of system requirements.
- Waterfall model of development is impractical
- An approach to development based on iterative specification and delivery is the only way to deliver software quickly.

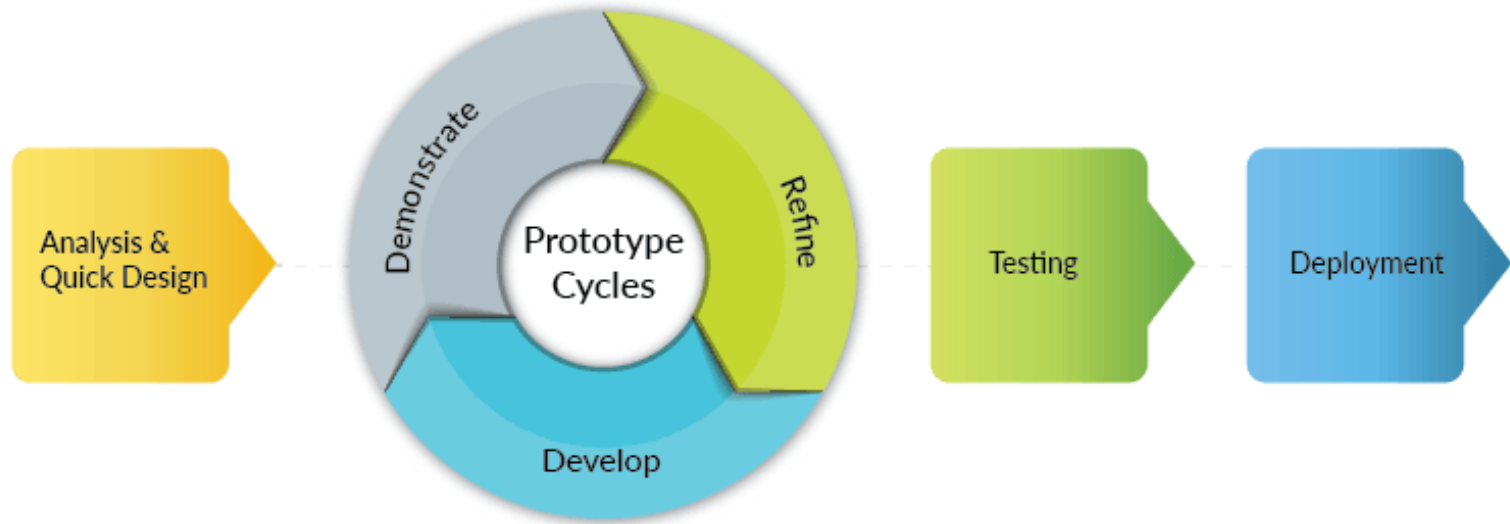
Characteristics of RAD processes

- Concurrent processes of specification, design and development.
- No detailed specification and minimized design documentation.
- The system is developed in a series of increments.
- End users evaluate each increment and make proposals for later increments.
- System user interfaces are usually developed using an interactive development system

Traditional Software Development



Rapid Application Development



Prototyping Software Development

What is a Prototype?

“A prototype is an initial version of a system used to demonstrate concepts and try out design options.”



What is a Prototyping?

“A process of building a model of the system to be developed.”

Uses of system prototypes

- ✓ To help customers and developers **understand the requirements for the system.**

- **Requirements elicitation**

Users can experiment with a prototype to see how the system supports their work.

- **Requirements validation**

The prototype can reveal errors and omissions in the requirements.

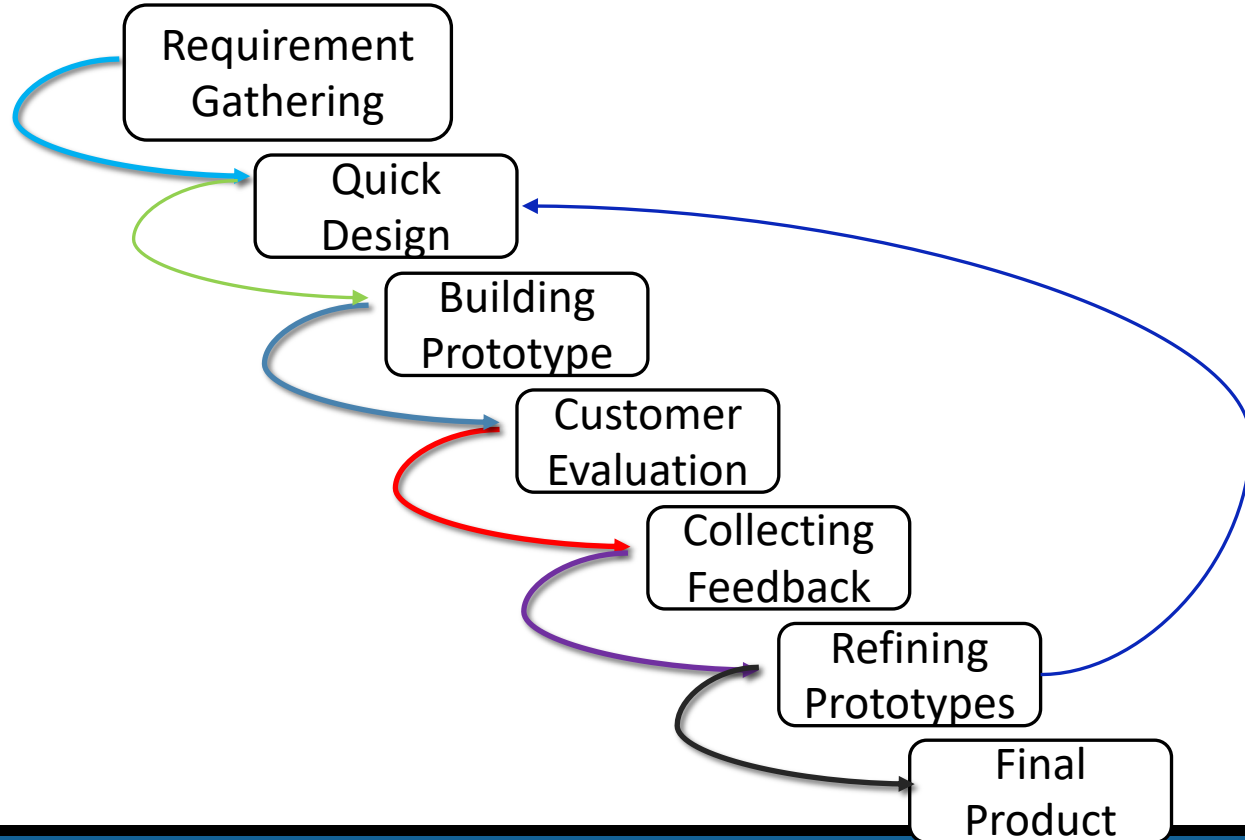
- ✓ Prototyping can be considered as a **risk reduction activity** which reduces requirements risks.

Purpose of Prototyping

The aim of prototyping is to **resolve uncertainty** about

- functional and user requirements
- operation sequences
- user support needs
- required representations
- “Look and Feel” of the interface
- appropriateness of the design

Stages in Prototyping



When to use Prototyping?

- When **requirements are unstable** / have to be clarified in the beginning of the project.
- When **quick delivery of the system is required**.
- Excellent for **designing good human computer interface** systems(GUI).
- **Technology is new or unknown** to the developers.
- Needs to **have a lot of interaction** with the end users.

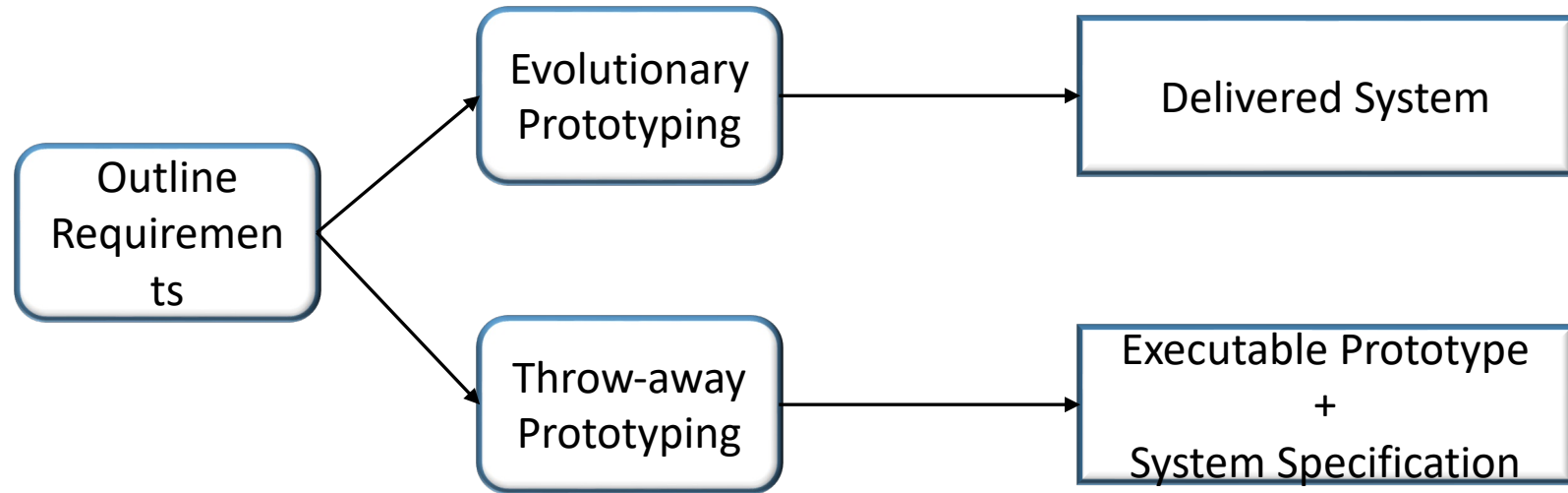
Advantages of Prototyping

- A closer match to users' real needs.
- Users can identify needed changes and refine real requirements
- A working system is available early in the process
- Serve as a basis for deriving a system specification
- The system can support user training and system testing
- Reduced development effort.

Disadvantages of Prototyping

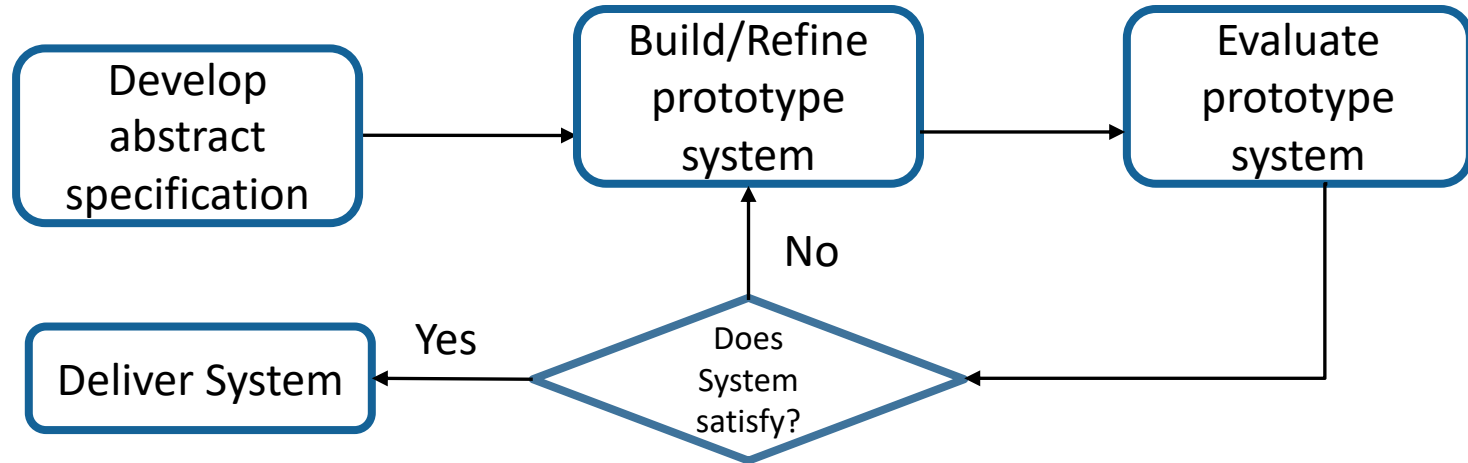
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- Too much involvement of client, is not always preferred by the developer.
- Too many changes can disturb the rhythm of the development team.
- Initial design decisions may be poor

Approaches in Prototyping



Evolutionary prototyping

- An **initial prototype** is produced and refined through a number of stages to the **final system**.
- The aim is **to deliver a working system to end-users**.
- The development **starts with those requirements which are best understood**.



Advantages of Evolutionary Prototyping

- Accelerated delivery of the system
- User engagement with the system
- The System is more likely to meet the user requirements.

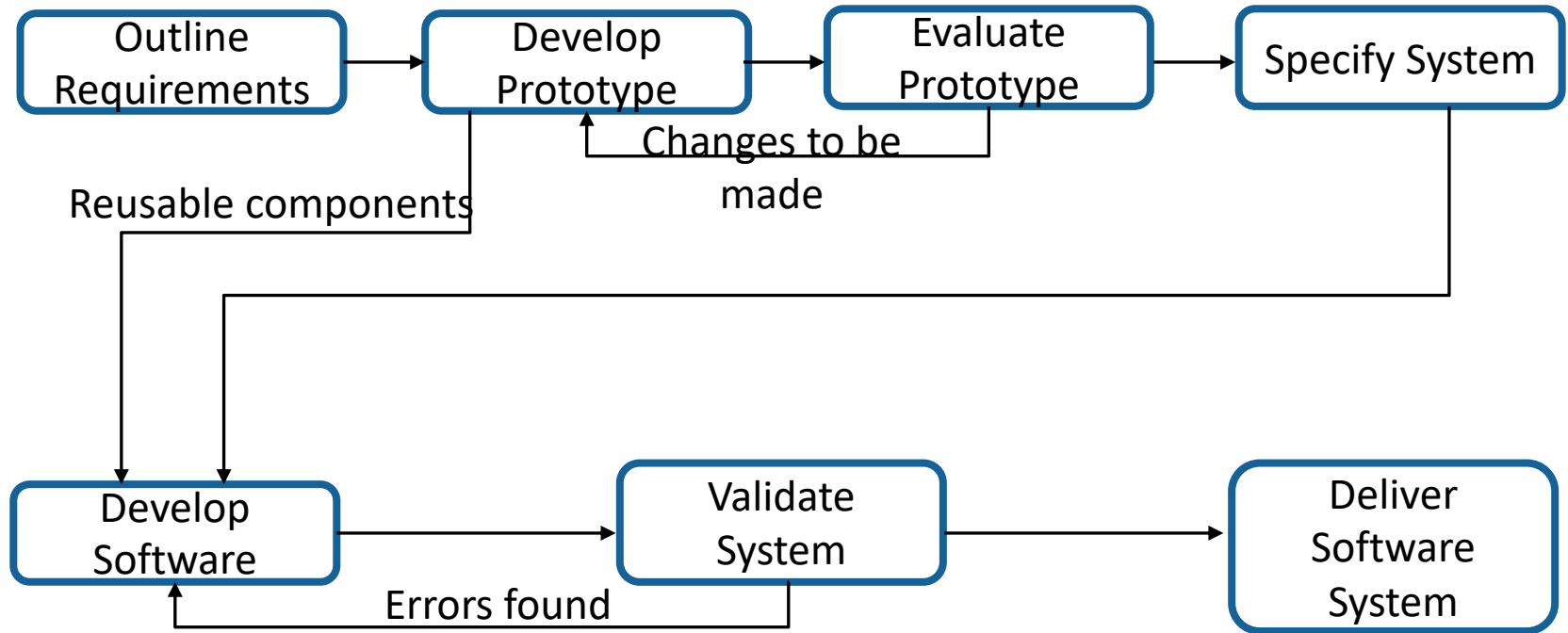
Disadvantages of Evolutionary Prototyping

- Requirements have to be analyzed quickly and carefully.
- Maintenance problems
- Continual change tends to corrupt system structure, so long-term maintenance is expensive.

Throw-away prototyping

- A prototype which is usually a practical implementation of the system is **produced to help discover requirements problems and then discarded.**
- The objective is to **ensure that the system requirements are validated and clearly understood.**
- **Starts with those requirements which are poorly understood.**
- The prototype is developed from an initial specification, **delivered for experiment then discarded.** The throw-away prototype **should NOT be** considered a **part of the final system.**

Throw-away prototyping...



Advantages of Throw-away Prototyping

- Aid understanding and reduce the risk of poorly defined requirements.
- Higher probability of satisfying all the requirements.
- Able to refine them early in the development of the software, if they get the quick feedbacks.
- Making changes early in the development lifecycle is extremely cost effective.

Disadvantages of Throw-away Prototyping

- Important features may have been left out of the prototype to similar rapid implementation.
- It may not be possible to prototype some of the most important parts of the system such as safety-critical functions within short time period.
- There is no specification for long-term maintenance
- The system will be poorly structured and difficult to maintain

Summary

- Why RAD?
- Characteristics of RAD
- Prototyping
 - Evolutionary Prototyping
 - Throw away Prototyping

Any Questions?

