FPT Education

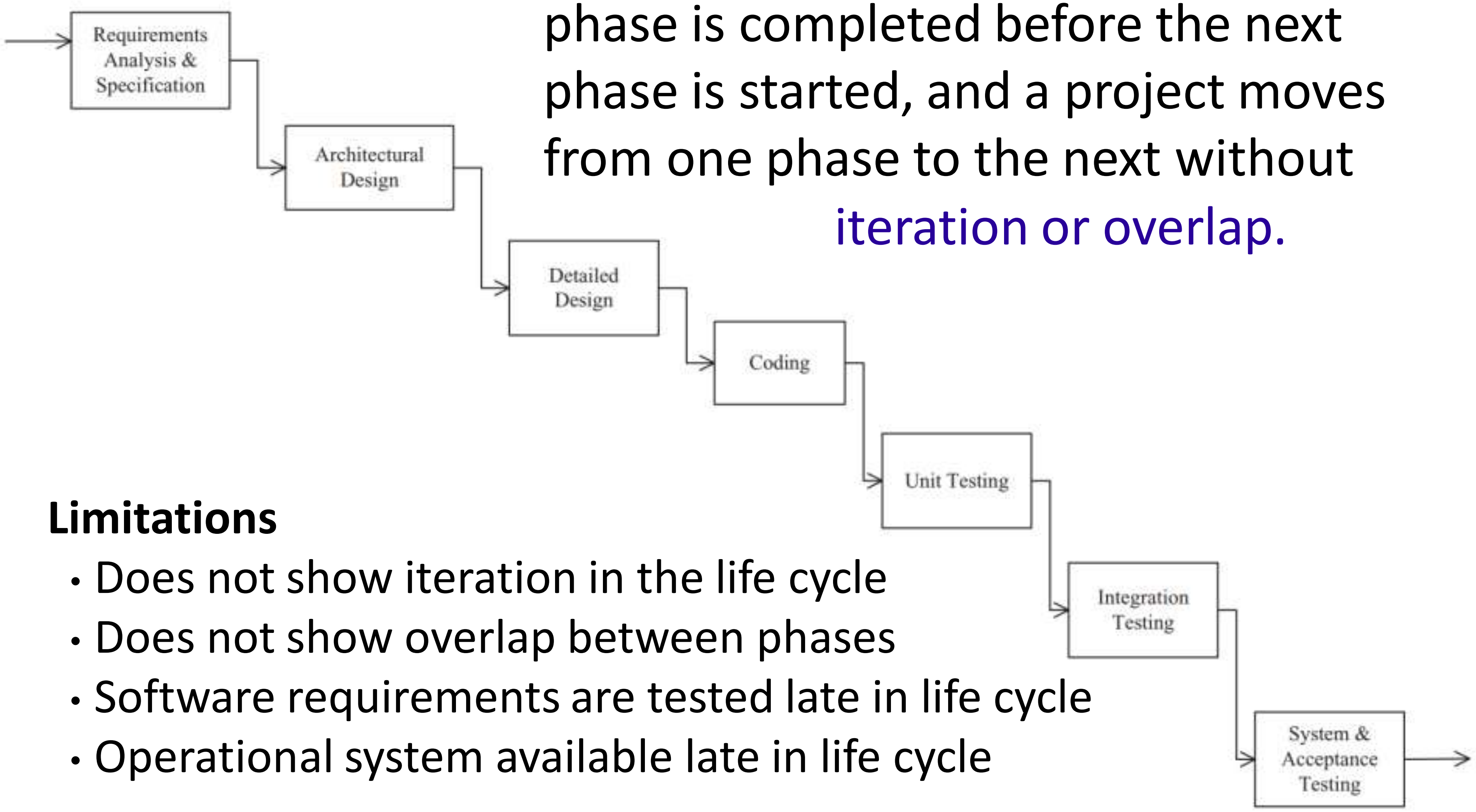
FPT UNIVERSITY

SOFTWARE DESIGN (SWD392)

CH03 – SOFTWARE LIFE CYCLE MODELS AND PROCESSES

Waterfall Model

Idealized process model in which each phase is completed before the next phase is started, and a project moves from one phase to the next without iteration or overlap.

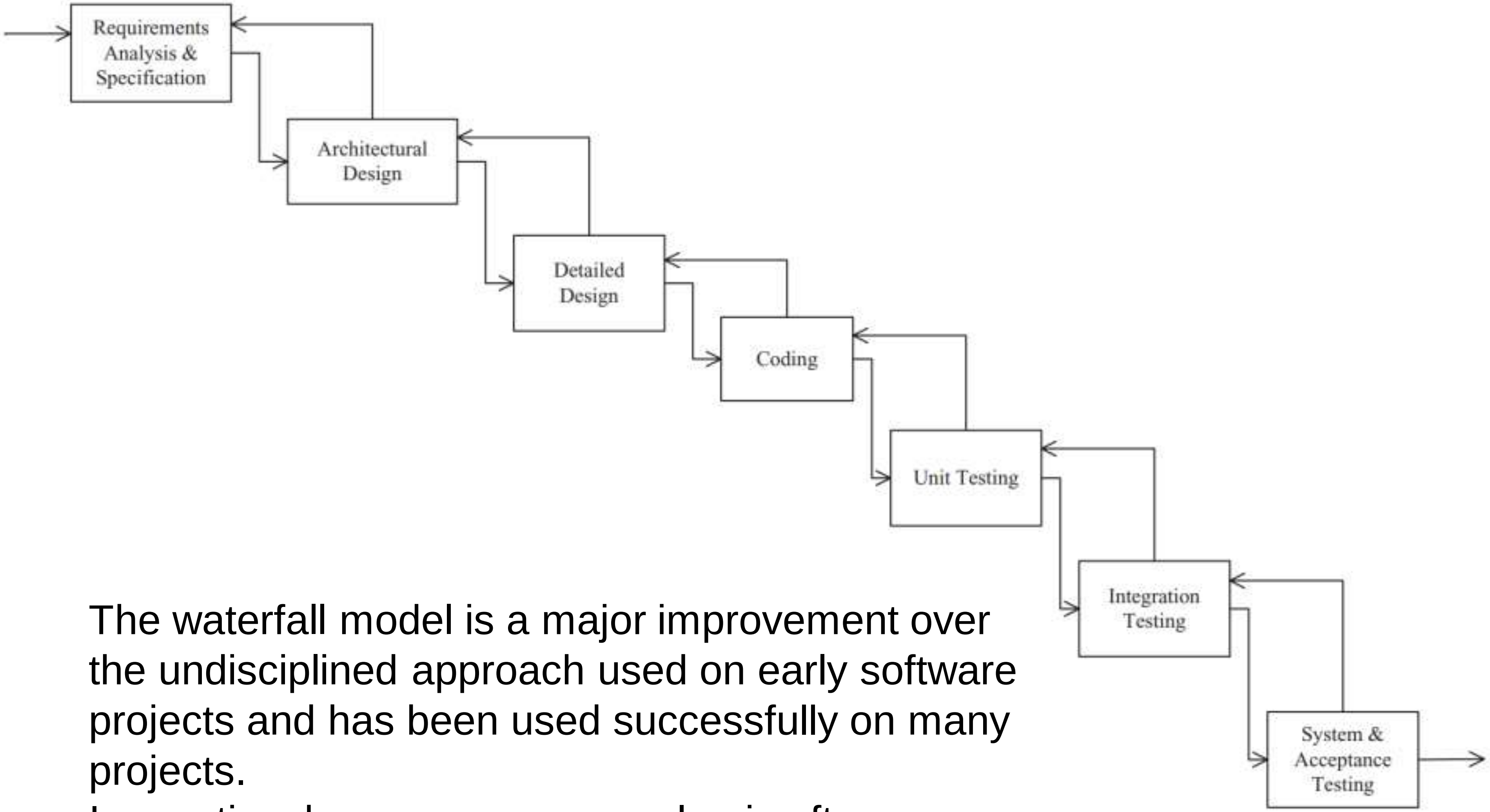


Limitations

- Does not show iteration in the life cycle
- Does not show overlap between phases
- Software requirements are tested late in life cycle
- Operational system available late in life cycle

Waterfall Model

Waterfall model with iteration between phases



The waterfall model is a major improvement over the undisciplined approach used on early software projects and has been used successfully on many projects.
In practice, however, some overlap is often necessary between successive phases of the life cycle, as well as some iteration between phases when errors are detected

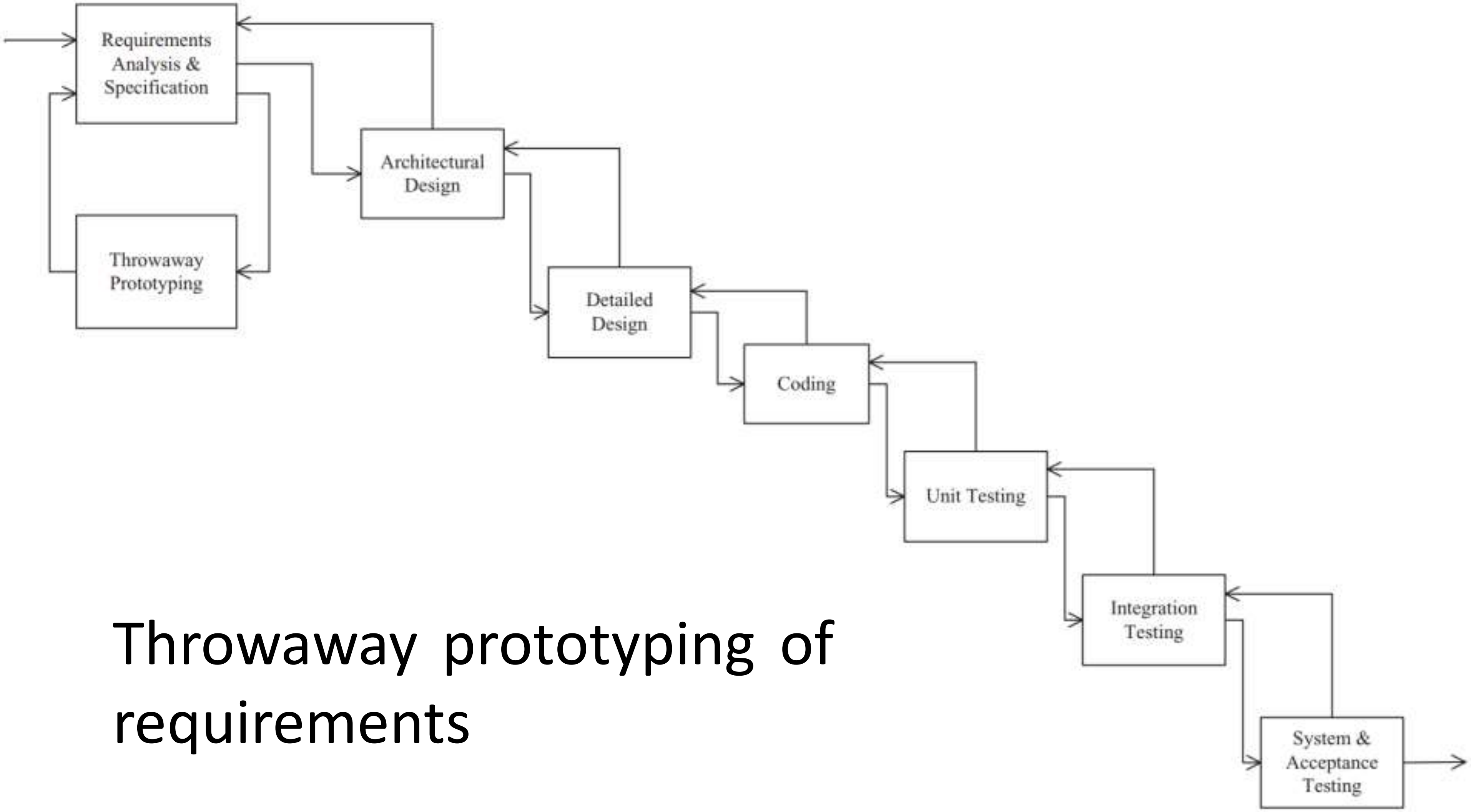
Throwaway Prototyping 1/3

Impact of Throwaway prototyping and its revolution

- Particularly useful for getting feedback on the User Interface
- Is developed after a preliminary requirements specification
- Is an effective solution to the problem of specifying the requirements for interactive information system
- Helps overcome the communication barrier that existed between the users and the developers
- Evolutionary prototyping approach is a form of incremental development in which the prototype evolves through several intermediate operational systems into the delivered system
- Evolutionary prototyping approach is to have a subset of the system working early, which is then gradually built on

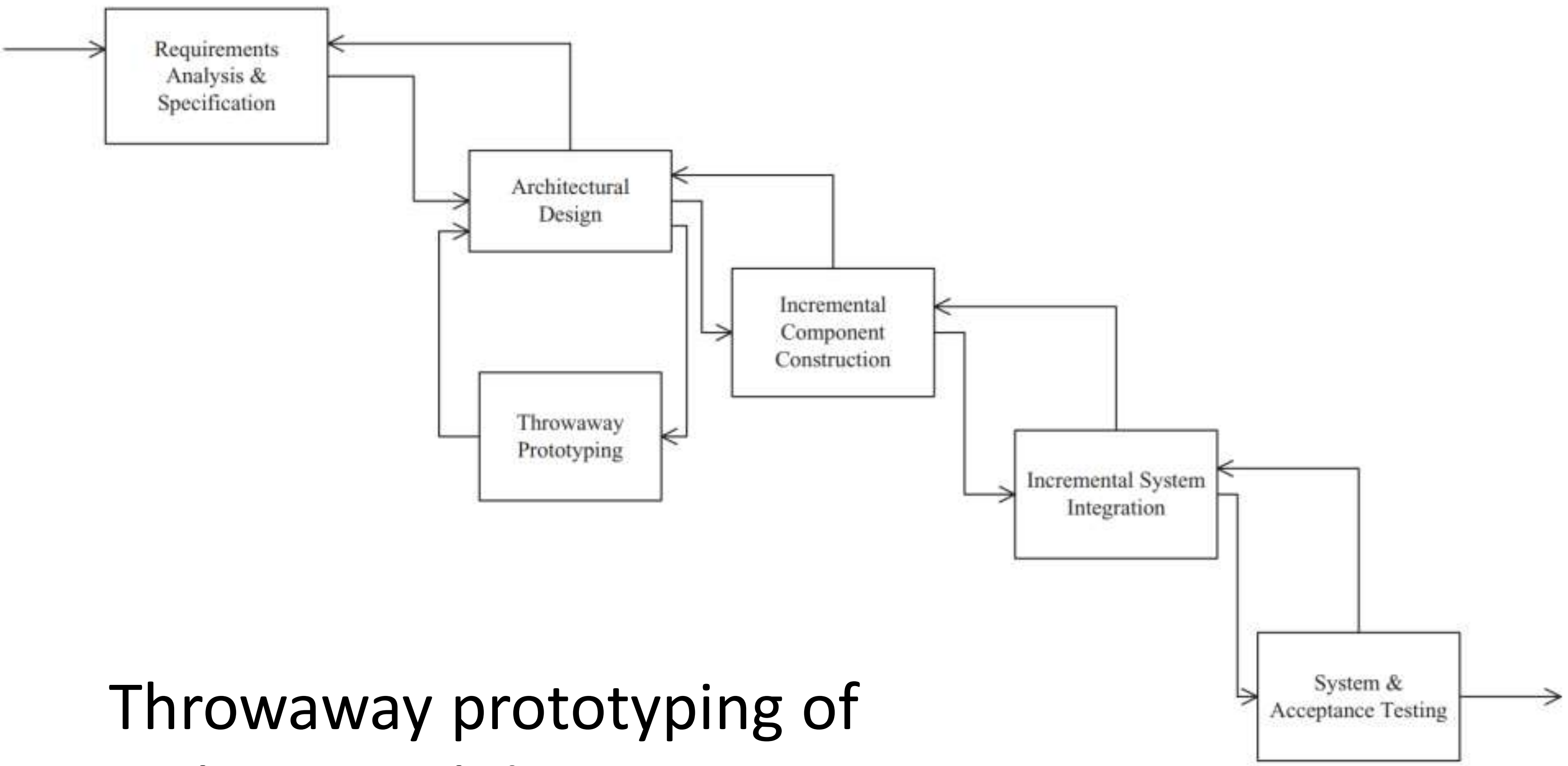
FPT Education
FPT UNIVERSITY

Throwaway Prototyping 2/3



Throwaway prototyping of requirements

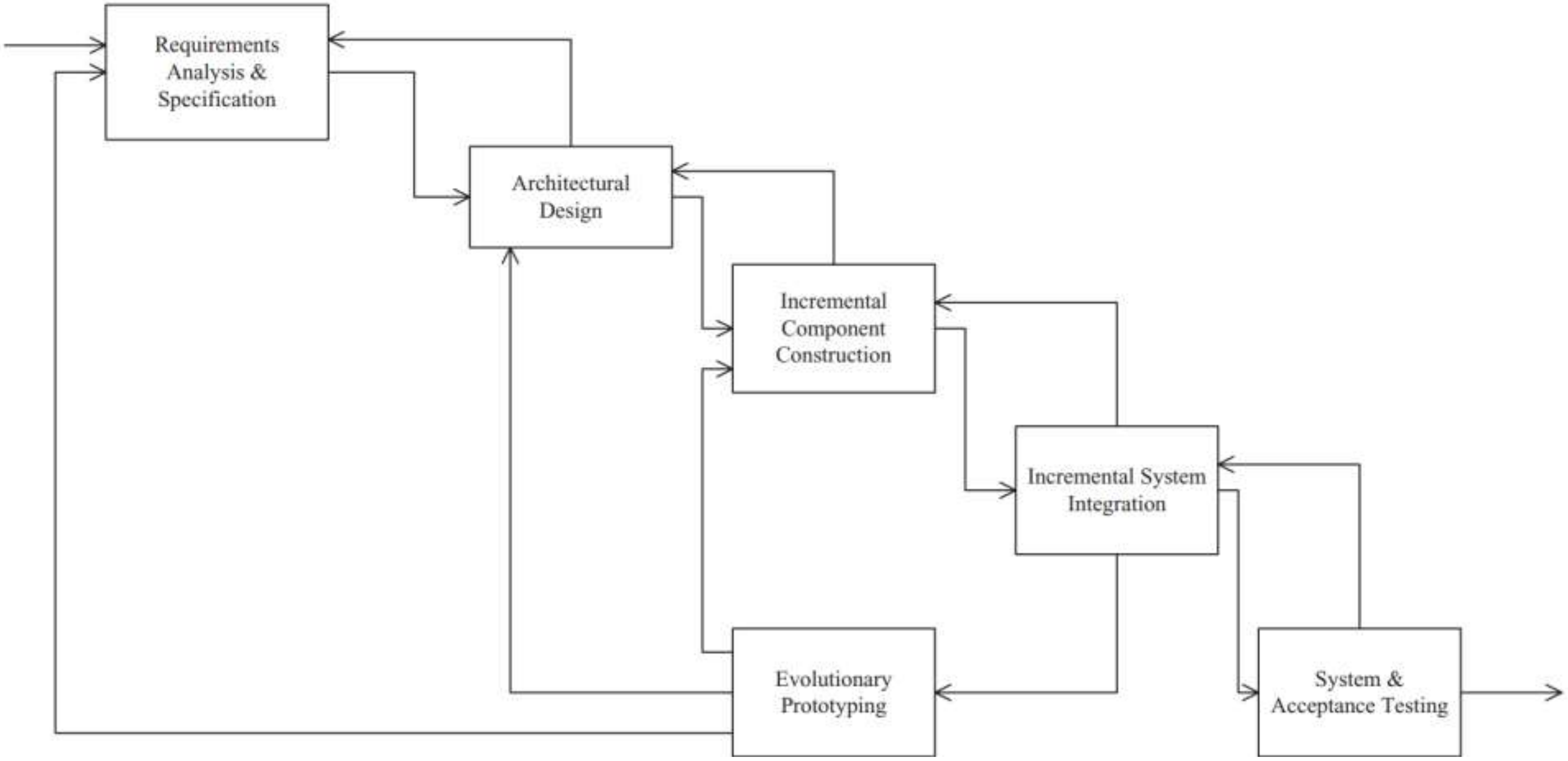
Throwaway Prototyping 3/3


Requirements Analysis & Specification
Architectural Design
Throwaway Prototyping
Incremental Component Construction
Incremental System Integration
System & Acceptance Testing

Throwaway prototyping of architectural design

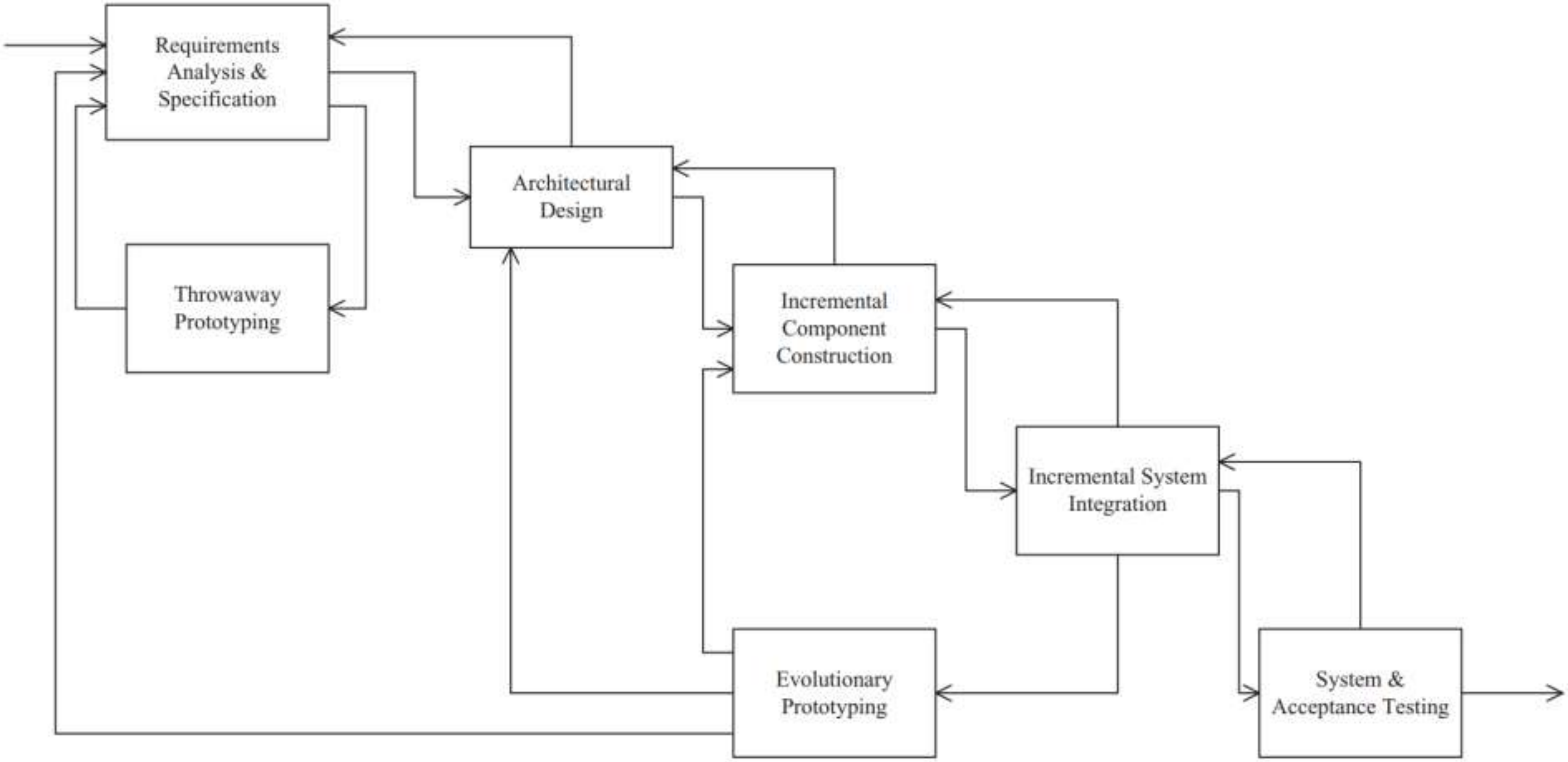
Incremental development model

Evolutionary Prototyping by Incremental Development


Requirements Analysis & Specification
Architectural Design
Incremental Component Construction
Incremental System Integration
Evolutionary Prototyping
System & Acceptance Testing

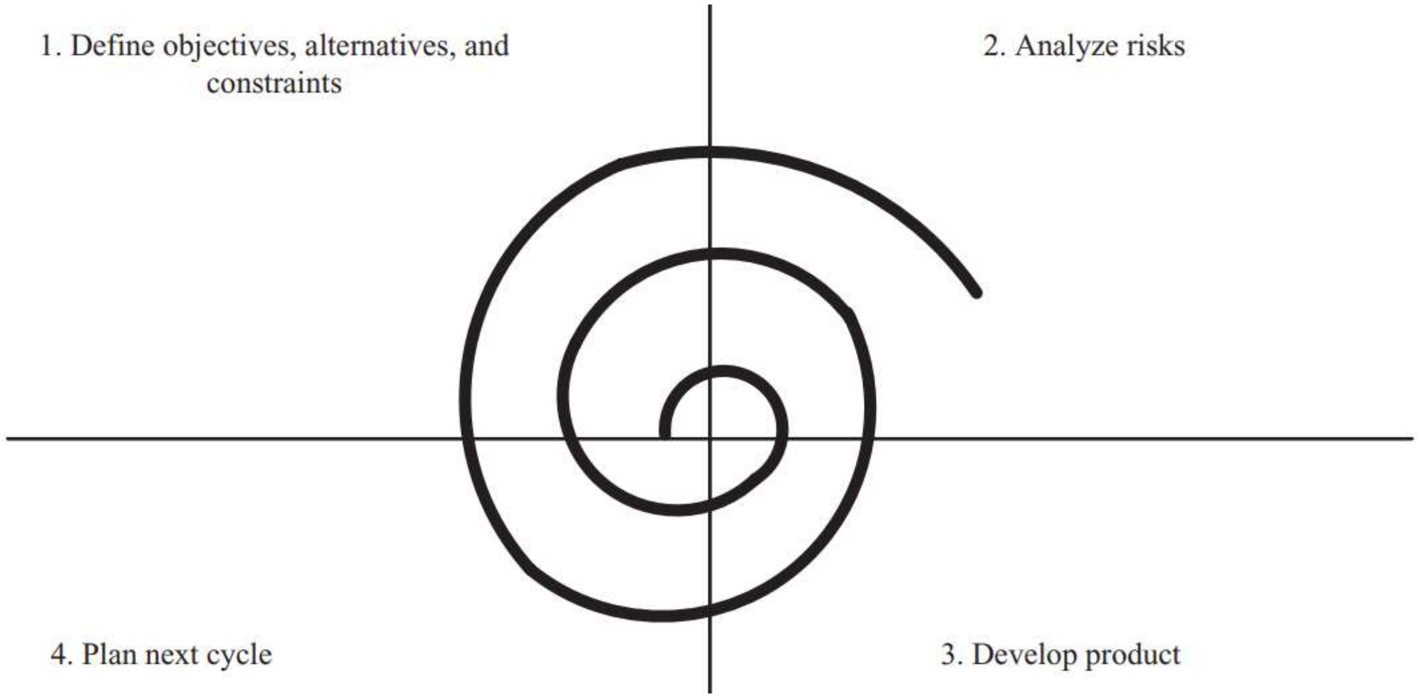
Incremental development model

Combine throwaway prototyping


Requirements Analysis & Specification
Throwaway Prototyping
Architectural Design
Incremental Component Construction
Incremental System Integration
Evolutionary Prototyping
System & Acceptance Testing

FPT Education
FPT UNIVERSITY

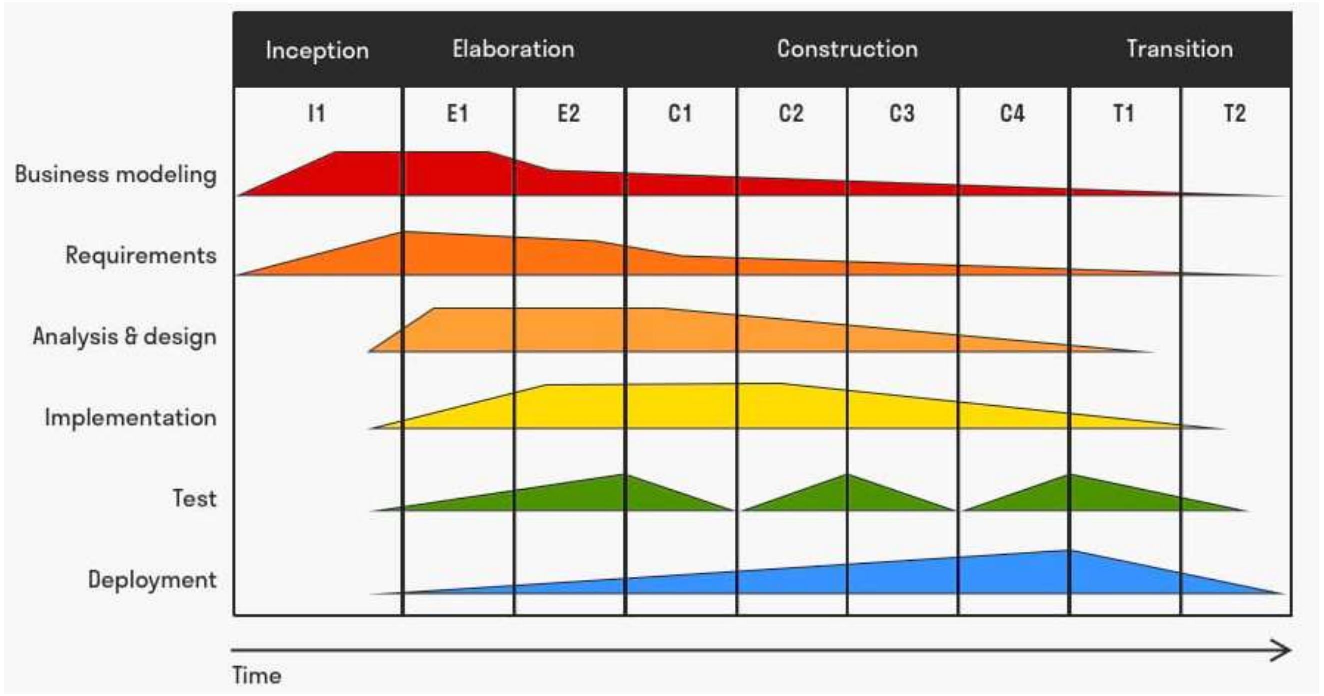
Spiral Process Model (SPM)



- Define objectives, alternatives, and constraints. Detailed planning for this cycle: identify goals and alternative approaches to achieving them.
- Analyze risks. Detailed assessment of current project risks; plan activities to be performed to alleviate these risks.
- Develop product. Work on developing product, such as requirements analysis, design, or coding.
- Plan next cycle. Assess progress made on this cycle and start planning for next cycle.

Rational Unified Process (RUP)

AKA Unified Software Development Process (USDP)



*Consists of five core workflows and **four iterative phases***

- *Inception: idea, concepts*
- *Elaboration: software architecture*
- *Construction: ready for release to the user community*
- *Transition: the software is turned over to the user community*

Design Verification & Validation

The goal of software validation is to ensure that the software development team “builds the right system,” that is, to ensure that the system conforms to the user’s needs.

The goal of software verification is to ensure that the software development team “builds the system right,” that is, to ensure that each phase of the software system is built according to the specification defined in the previous phase.

Design Verification & Validation

Verification & Validation Activities

Software Quality Assurance – activities to ensure the quality of the software product

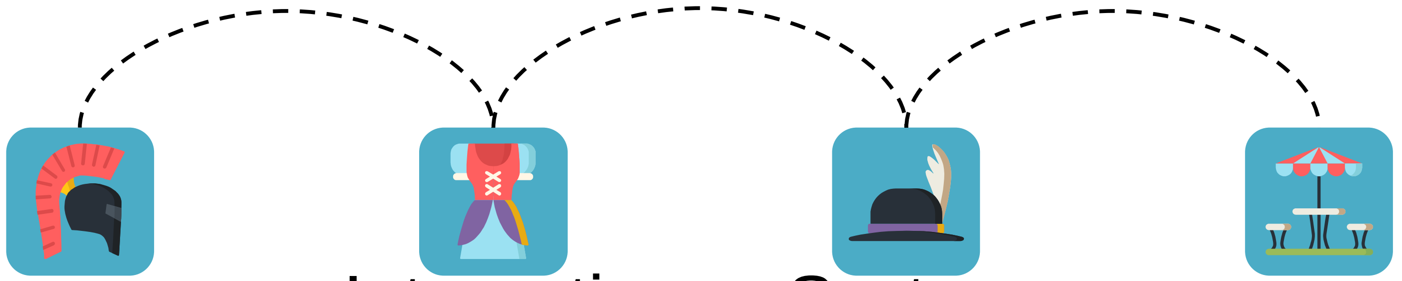
- Throwaway prototyping: validation of the system (before it is developed) against the user requirements
- Software technical reviews can help considerably with software verification and validation. In software verification, it is important to ensure that the design conforms to the software requirements specification

Performance Analysis of Software Designs

- Analyzing the performance of a software design before implementation is necessary to estimate whether the design will meet its performance goals.
- If potential performance problems can be detected early in the life cycle, steps can be taken to overcome them

Software Lifecycle Activities

- Requirements Analysis & Specification: BRD/URD => SRS
- Architectural Design: define overall system structures
- Detailed Design: define algorithms, internal structures,..
- Coding: each component is coded in the programming language selected for the project
- Software Testing



Unit Test

- Tests on individual components
- Uses test-coverage criteria

**Integratio
n**

- Involves combining tested components into progressively complex grouping

System Test

- Functional testing
- Load testing
- Stress testing
- Volume testing

AT

- Carries out by the end users

Q
A