

PROCESS MODEL

Lecture 4

What is a Process Model ?

*It is a **description** of*

- i) what tasks need to be performed in*
- ii) what sequence under*
- iii) what conditions by*
- iv) whom to*
achieve the “desired results.”

Plan-driven and Agile Processes(Development Approach)

- **Plan-driven processes are processes where all of the process activities are planned in advance and progress is measured against this plan.**
- **In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements.**

In practice, most practical processes include elements of both plan-driven and agile approaches.
There are no right or wrong software processes.

Why Have A Process Model?

- Provide “**guidance**” for a systematic coordination and controlling of
 - a) **the tasks** and of
 - b) **the personnel** who perform the tasks

Note the key words: coordination/control, tasks, people

Extending the “Simple” Process

As projects got larger and more complex.

- Needed to **test more functionalities**
- Needed to **clarify and stabilize the requirements**
- Needed to **design more carefully**
- Needed to **use more existing software & tools**
 - Database
 - Network
 - Code control
- Needed **more people** to be involved

Resulting in more tasks and more people

Effectiveness of using Correct Process Model

By changing the process model, we can improve :

- *Development speed (time to market)*
- *Product quality*
- *Project visibility*
- *Administrative overhead*
- *Risk exposure*
- *Customer relations, etc.*

Normally, a process model covers the entire lifetime of a product.
From *birth of a commercial idea* to *final installation of last release*

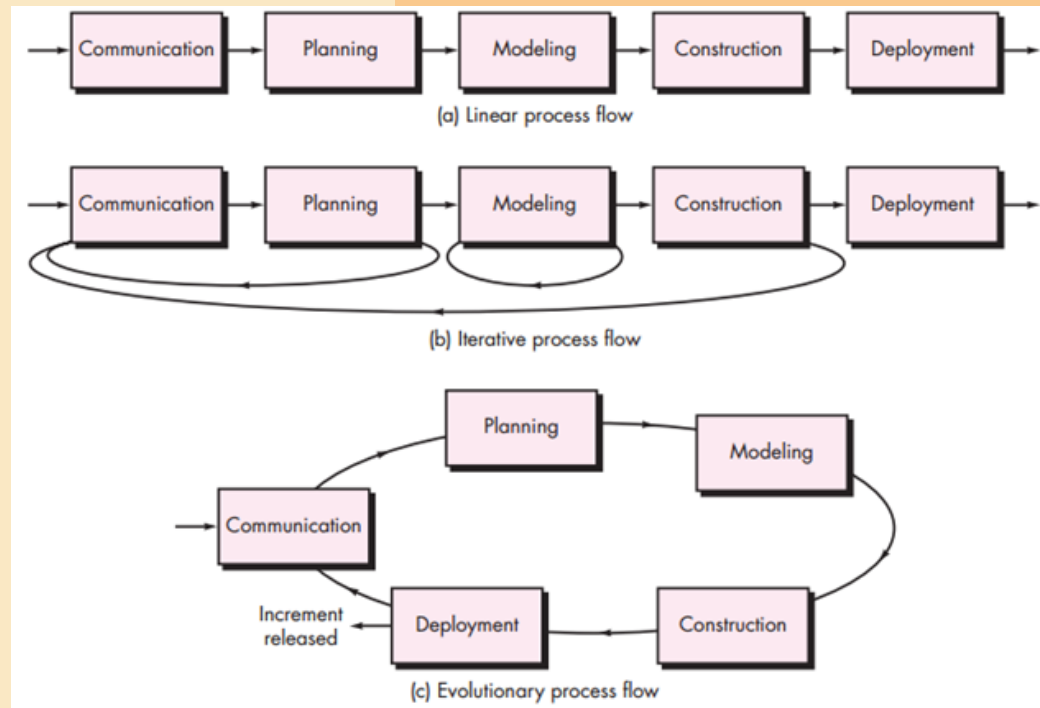
Common Activities of Process Model

Many different software processes but all involve:

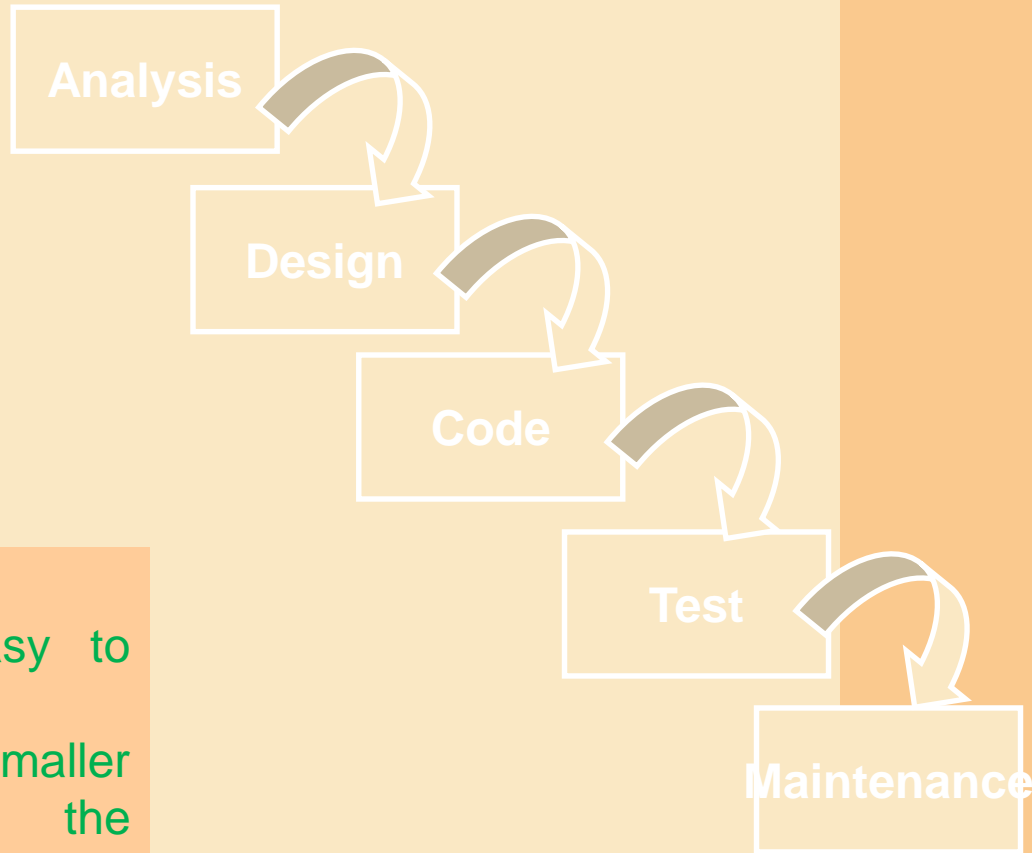
- **Specification** – defining what the system should do;
- **Design and implementation** – defining the organization of the system and implementing the system;
- **Validation** – checking that it does what the customer wants;
- **Evolution** – changing the system in response to changing customer needs.

Generic process model

- **Linear process flow**
- **Iterative process flow**
- **Incremental process flow**



Waterfall Model



Pros

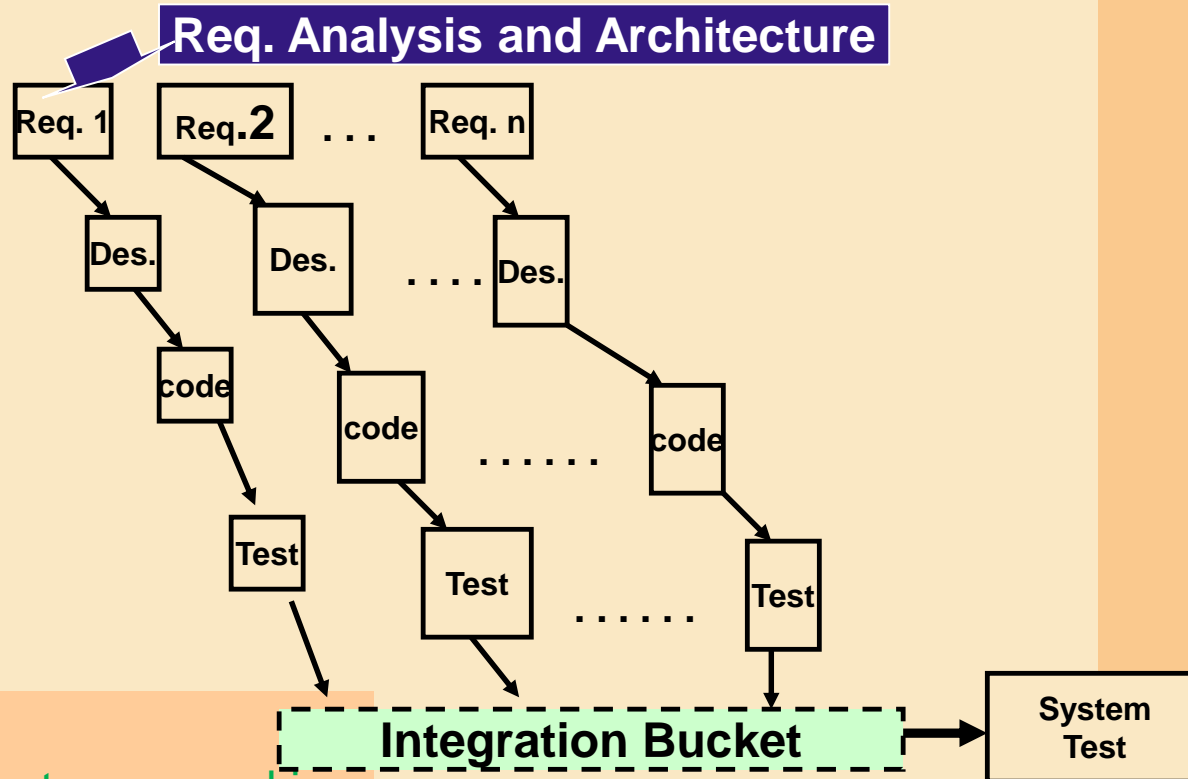
- Simple and easy to understand.
- Works well for smaller projects where the requirements are well-understood.

Cons

- No working product
- Inflexible
- Poor model for large projects.

1. Requirements must be specified.
2. Four main tasks must be completed in sequence: Analysis, design, code, and test, followed by integration of the system.
3. Output of one stage feeds into the next stage in sequence

Incremental Model



1. Each “major requirement/item” is further developed separately through the same sequence of : requirement, design, code, and unit test.

2. As the developed pieces are completed, they are continuously merged and integrated into a common bucket for integrated system test

Pros

- Generates working software quickly.
- More flexible
- Less costly
- easier to test and debug.
- customer can respond to each built.

Cons

- Needs good planning and design.
- Needs a clear and complete definition of the whole system

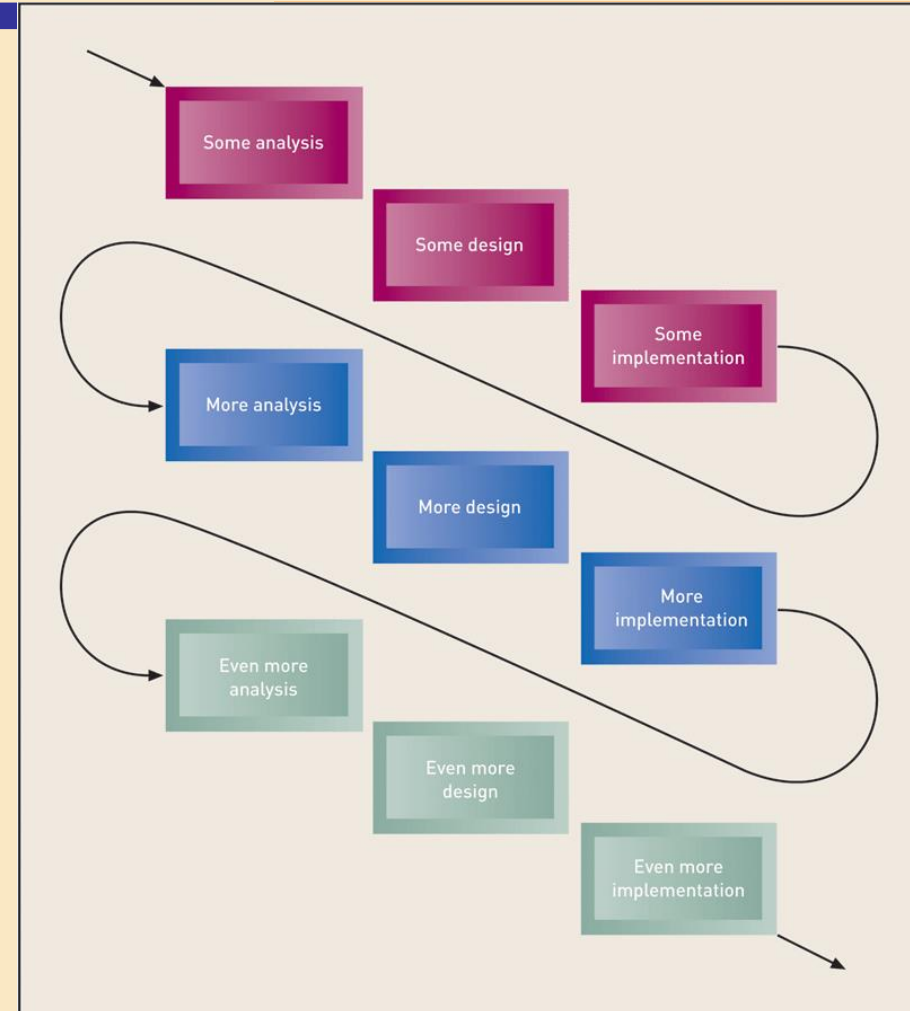
Iteration of System Development Activities

Problems

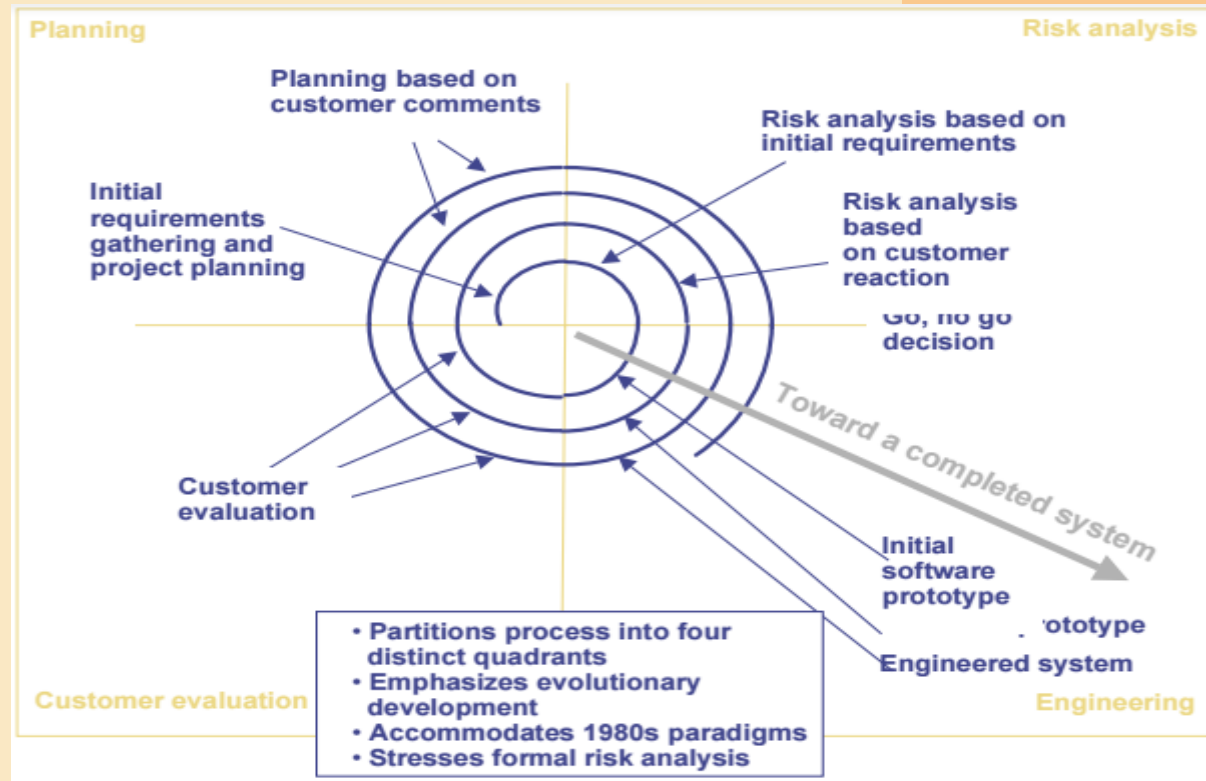
- Lack of process visibility
- Systems are often poorly structured

Applicability

- For small or medium-size interactive systems
- For parts of large systems
- For short-lifetime systems



Iteration Model



AGILE MODEL

Definition

The requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.

Break tasks into small increments with minimal planning and do not directly involve long-term planning.

Iterations are short time frames (time boxes) that typically last from one to four weeks.

Agile Development Characteristics

- **Program specification, design and implementation are inter-leaved**
- **The system is developed as a series of versions or increments with stakeholders involved in version specification and evaluation**
- **Frequent delivery of new versions for evaluation**
- **Extensive tool support (e.g. automated testing tools) used to support development.**
- **Minimal documentation – focus on working code**

Plan-driven and Agile Development

- **Plan-driven development**

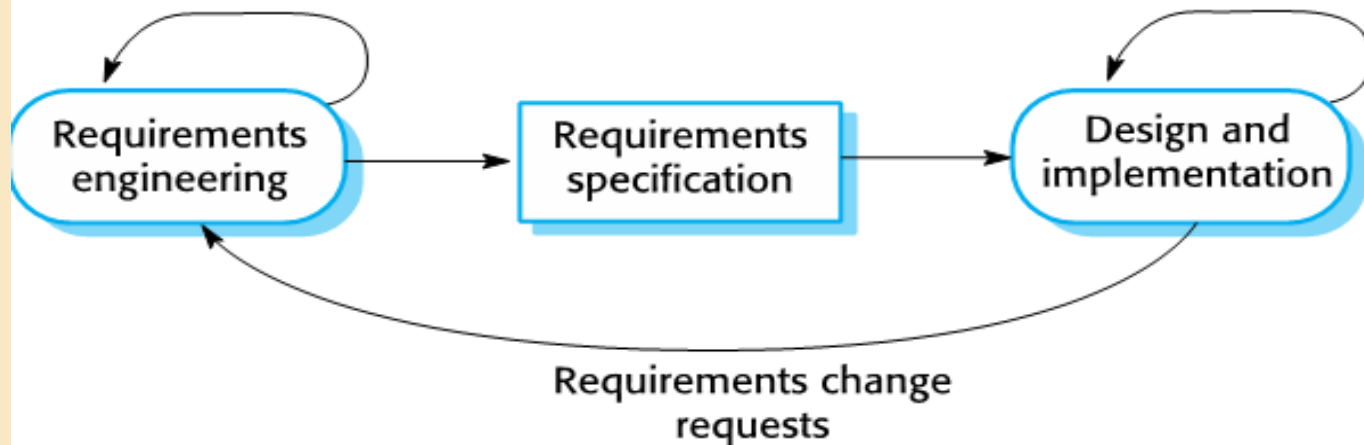
- A plan-driven approach to software engineering is based around separate development stages with the outputs to be produced at each of these stages planned in advance.
- Not necessarily waterfall model – plan-driven, incremental development is possible
- Iteration occurs within activities.

- **Agile development**

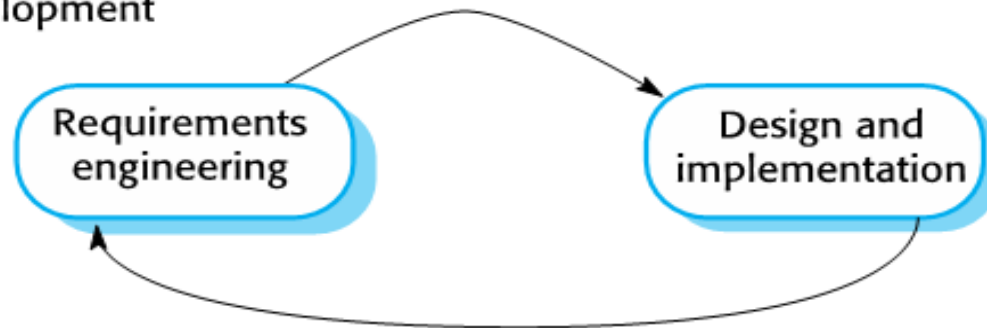
- Specification, design, implementation and testing are inter-leaved and the outputs from the development process are decided through a process of negotiation during the software development process.

Plan-driven and Agile Development

Plan-based development



Agile development



Agile

- Each iteration involves a *cross functional* team working in all functions:
 - Planning, and requirements analysis
 - Design, and coding
 - unit testing and acceptance testing
- At the end of the iteration a **working product** is demonstrated to stakeholders.

Characteristics of Agile Mode

Iterative & Incremental

Test Driven Development

People Oriented

Ligtweight

UNIFIED PROCESSING

Unified Processing

What is an **Unified Processing**?

A process model that was created 1997 to give a framework for Object-oriented Software Engineering

Iterative, incremental model to adapt to specific project needs

The Unified Process is an adaptable methodology

The Unified Process is a modeling technique

The System Development Life Cycle

What are Characteristics of the Unified Process?

Object Oriented

Use Case Driven

Architecture Centric

Iteration & Increment

Focus core architecture in the early iterations

In earliest iterations, get high valued requirements

View of the whole design with the important characteristics made more visible

Expressed with class diagram

- Utilizes object oriented technologies.
- Classes are extracted during OOA and designed during OOD.

- Utilizes use case model to describe complete functionality of the system

- Iterations are time boxed (i.e. fixed in length)

- Best iteration length (2-6 weeks)

The System Development Life Cycle

What are the phases of UP?



Inception

Elaboration

Construction

Transition

Inception

- A vision of the product is created. Questions discussed are:
- What is the product supposed to do?
- Why should my organization embark on a project to build this particular product?
- Does my organization have the resources to build this product?
- Is it feasible to do so?
- How much will this product cost and how much will it bring in?
- What will be the duration of the project?
- Risk analysis is performed.
- Decision whether to go ahead with the project or not is taken.

Elaboration

- System to be built is analyzed in detail.
- Use cases used to document the requirements.
Main aim: Get the core architecture and as many use cases as possible.
- The core architecture is coded, verified with user and baselined.
- Other high-risk requirements are identified and coded.
- A project plan is drawn in this phase, resources are allocated and a schedule is planned.
- UML diagrams are used to model the system under design.

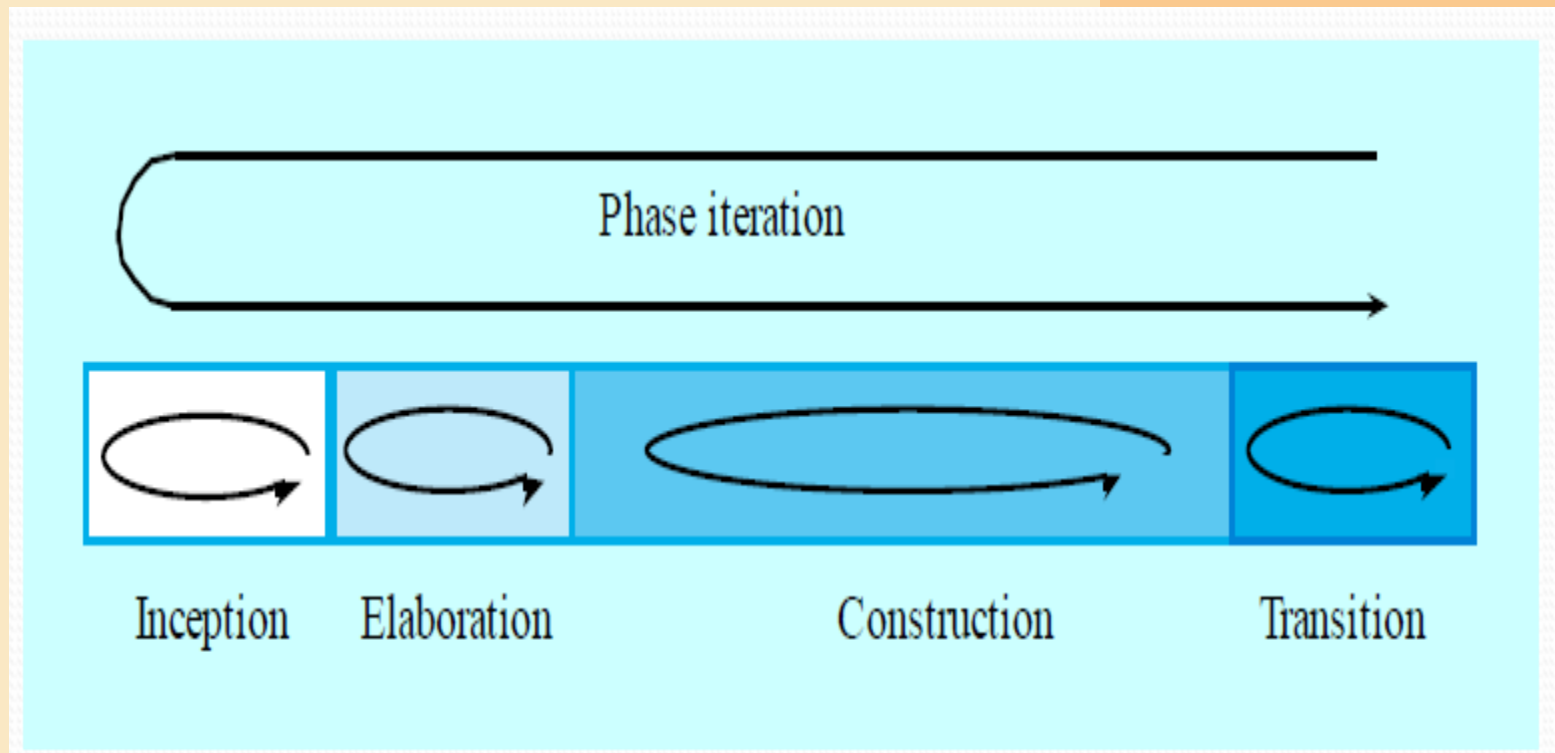
Construction

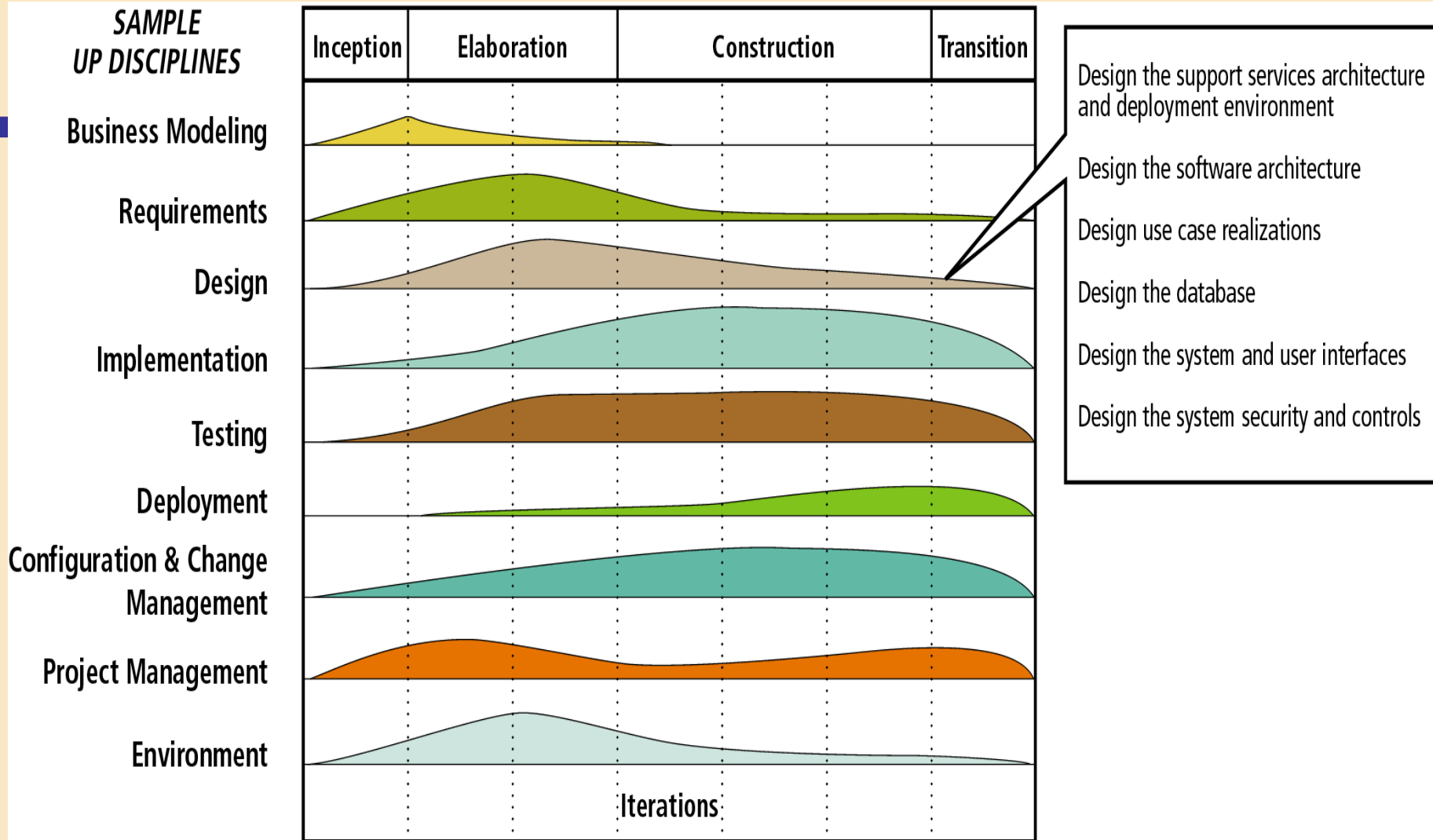
- Remaining use cases are implemented.
- If any new use cases are discovered, they are implemented.
- Test cases are written, actual tests are carried out and a test report is prepared.
- Documentation for the system as well as guides for the users are written.

Transition

- System is installed in its environment and beta-tested.
- Feedback is received and System is refined and tuned to adapt in response to the feedback.
- It also includes activities like marketing of the product and training of users.

Unified Process Model





Design Activities in the UP Life Cycle

Repeated verification of quality

- The UP stresses on the repeated verification of quality.
- Here, quality refers to both, the quality of the software to be developed as well as the quality of the process itself.
- At the end of each iteration the team must produce executable software.
- This achievement indicates the success of the iteration.
- The software is tested and verified right from the start unlike the traditional SDLC where Quality Analysis is done basically at the end.

The Unified Process (UP) (continued)

- **Reinforces six best practices**
 - **Develop iteratively**
 - **Define and manage system requirements**
 - **Use component architectures**
 - **Create visual models**
 - **Verify quality**
 - **Control changes**

Phase Deliverables

Inception Phase	Elaboration Phase	Construction Phase	Transition Phase
<ul style="list-style-type: none">• The initial version of the domain model• The initial version of the business model• The initial version of the requirements artifacts• A preliminary version of the analysis artifacts• A preliminary version of the architecture• The initial list of risks• The initial ordering of the use cases• The plan for the elaboration phase• The initial version of the business case	<ul style="list-style-type: none">• The completed domain model• The completed business model• The completed requirements artifacts• The completed analysis artifacts• An updated version of the architecture• An updated list of risks• The project management plan (for the rest of the project)• The completed business case	<ul style="list-style-type: none">• The initial user manual and other manuals, as appropriate• All the artifacts (beta release versions)• The completed architecture• The updated risk list• The project management plan (for the remainder of the project)• If necessary, the updated business case	<ul style="list-style-type: none">• All the artifacts (final versions)• The completed manuals

WORKFLOWS

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.
Test	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 managed changes to the system
Project management	This supporting workflow manages the system development
Environment	This workflow is concerned with making appropriate software tools available to the software development team.