

SE 3140

DESIGN AND MODELLING

M. Mangong Clement

SE 3140

DESIGN AND MODELLING

By M. Mangong C. Fosah

mangong.clement@ictuniversity.org

(+237) 653 519 879

M. Mangong Clement

Chapter 1:

Introduction to Software Design and Modelling

Mangong C. Fosah
ICT Univerisity

Chapter 1: Topics Overview

- **Understanding basic Modelling terms**
- **The importance of Modelling**
- **The Need for Model**
- **Impact of software fault**
- **Assignment**

Understanding basic terms

➤ **What is modelling?**

- Used in science and engineering to provide abstractions of a system at some level of precision and details.

□ Understanding basic terms Contd...

➤ Modeling

- It is used in science and engineering to provide abstractions of a system at some level of precision and details.



b) The great pyramid of Egypt



□ Understanding basic terms Contd...

➤ Modeling

- The model is then analyzed in order to obtain a better understanding of the system being developed.



b) The great pyramid of Egypt



□ Understanding basic terms Contd...

- **Why Modelling? (None software example 1)**
 - To architect a dog house
 - Can be done by one person. It involves a simple process, performed with basic tools.
 - In few hours you end with a house for dog and can do with no one else's help.
 - Modeling effort is minimal or sometimes unnecessary.




□ Understanding basic terms *Contd...*

- **Why Modelling? (None software example 2)**
 - If you want to build a house for your family, is it possible?
 - It requires detailed planning and some sketches etc.



□ Understanding basic terms Contd...

➤ What is software modeling?

- Is it expressing a scientific theory or algorithm in software ? 
- Is it larger than an algorithm or a single method. 
- Does it address the entire software design including **interfaces**, **interactions** with other software, and all the software methods. 

□ Understanding basic terms Contd...

- **What is software modeling?**
 - It is the designing of software application before coding (*Object Modeling Group – OMG*)
 - In model-based software design, software modeling is used as an essential part of software development.
 - Models are built and analyzed before system implementation.

□ Understanding basic terms Contd...

➤ **What is software model?**

- A model is a simplification with a purpose.
- It uses a precisely define notation to describe and simplify a complex and interesting structure, phenomenon, or relationship.
- Abstractions of the reality.
- ✓ Abstraction are simplification that ignore irrelevant details, only represent the relevant details.

□ Understanding basic terms Contd...

➤ What is software model? (1)

- Abstractions that allows representation of various layers of complex information regarding a software.
- Anything that uses abstraction to capture pieces of a software.
- A way of expressing a software design
- The models are used to create the software architecture.

Understanding basic terms *Contd...*

➤ **What is software model? (2)**

- These are ways of expressing a software design.
- A sort of abstract language or pictures are used to express the software design.
- For object-oriented software, an object modeling language such as UML is used to develop and express the software design.

□ Understanding basic terms Contd...

➤ **What is software model ? (3)**

- One of the reasons to start a software by building models is actually built to find out if something will work.
- Software model helps developers visualize, communicate and validate a system before significant amounts of money are spent.
- ✓ This means, verifying and validating the model is very important.

Understanding basic terms *Contd...*

➤ **What is software model ? (4)**

- Unified Modeling Language (UML) is a modeling tool used to build models especially object oriented models.
- A modeling language is use to develop the design not just to capture the design after its complete.

□ Understanding basic terms *Contd...*

➤ **Unified Modeling Language (UML)**

- The UML allows the designer to try different designs and decide which will be best for the final solution.
- A standard graphical modeling language that helps to develop, understand and communicate the different views.

□ Understanding basic terms Contd...

➤ **Unified Modeling Language (UML) (1)**

- Just as C++ is the programming language used for implementing your software in programming, so is UML used for building a software design and implemented in a programming language that is object-oriented.
- That implies that UML is considered to be an object-oriented modeling language.

□ Understanding basic terms Contd...

➤ Unified Modeling Language (UML)(2)

- The UML allows the designer to try different designs and decide which will be best for the final solution.
- A standardized graphical notation for describing object-oriented models.

Understanding basic terms *Contd...*

➤ **Why UML?**

- **It is methodology-independent**
- ✓ It is a notation for describing the results of an object-oriented and design developed via the methodology of choice.
- ✓ It need to be used together with an object-oriented analysis and design method.

Understanding basic terms Contd...

➤ Why UML?

- Permits you to specify the structure or behavior of a system.
- Helps you visualize a system.
- Provide template that guides you in constructing a system.
- Helps to understand complex system part by part.
- Document the decisions that you have made.

Understanding basic terms *Contd...*

UML Tools

- It is controlled by the Object Management group (OMG)
 - Version 1.x
 - Version 2.x most UML tools support this version for its has stronger semantics

□ Understanding basic terms Contd...

➤ UML Tools

- There are several tools that you can use to develop your UML design.

Examples



□ Understanding basic terms Contd...

➤ UML Tools

- Provides three basic types of models

1. Use case models (Requirement oriented models)
2. Static models (complementary set of design models)
3. Dynamic models

The three models represents different viewpoints of the model.

□ Understanding basic terms Contd...

➤ UML Tools

- Use case Models
 - ✓ Captures the requirement of a system
 - ✓ Captures outside view of a system
 - ✓ Identify the external and internal factors influencing the system
 - ✓ Show the interaction among the requirements
 - ✓ It is the overall big picture of the requirements and what the system has to do.

□ Understanding basic terms Contd...

➤ UML Tools

■ Static Models

It tells us what information we have in the system and what a sub-system needs to know about another sub-system. Captures basic structure and static elements such as:

- what type of classes do we have in the system
- what information do we have in the classes
- what do the classes need to know about each other etc.

□ Understanding basic terms Contd...

➤ UML Tools

■ Dynamic Models

It tells us how things work in response to generating events coming into the system.

- Captures behavioral elements (how the system works). It will use the classes defined in the static models to create instances of those classes called object.

- It will show how the objects will communicate back and forth to satisfy the use cases

□ Understanding basic terms Contd...

➤ Basic characteristics of software models

❖ Simplification

- Models of software are far less complex;
- ✓ More accessible than the actual code and components that makes up the final system.
- It's easier for a developer to work to built, extern and evaluate a visual model than to work directly in the code.

□ Understanding basic terms Contd...

➤ Basic characteristics of software models

❖ Simplification(1)

- ✓ When decisions are made in code, they turn to stay made but if made at the level of the model, the code can easily be maintain.
- ✓ With modeling, and especially with a visual modeling tool, decisions can be made and revised quickly and efficiently.

□ Understanding basic terms Contd...

➤ **Basic characteristics of software models(1)**

❖ **Varying perspective**

- A single model of a software system can describe the system from different perspectives.
- ✓ One view may show how major parts of the system interact and cooperate.
- ✓ Other views might zoom in a particular subsystem part.

□ Understanding basic terms Contd...

➤ Basic characteristics of software models(2)

❖ Common notation

- A precise software model in a common notation allows developers to combine their efforts to work in parallel.
- As long as each contribution fits the model, the parts can be combined into the final system.
- Modern manufacturing uses this technique to reduce cost of software production schedule.

□ Understanding basic terms *Contd...*

➤ Principles of UML modeling

1. The choice of model is important.
2. Every model may be expressed at different levels of precision.
3. The best models are connected to reality.
4. No single model is sufficient.

□ Understanding basic terms Contd...

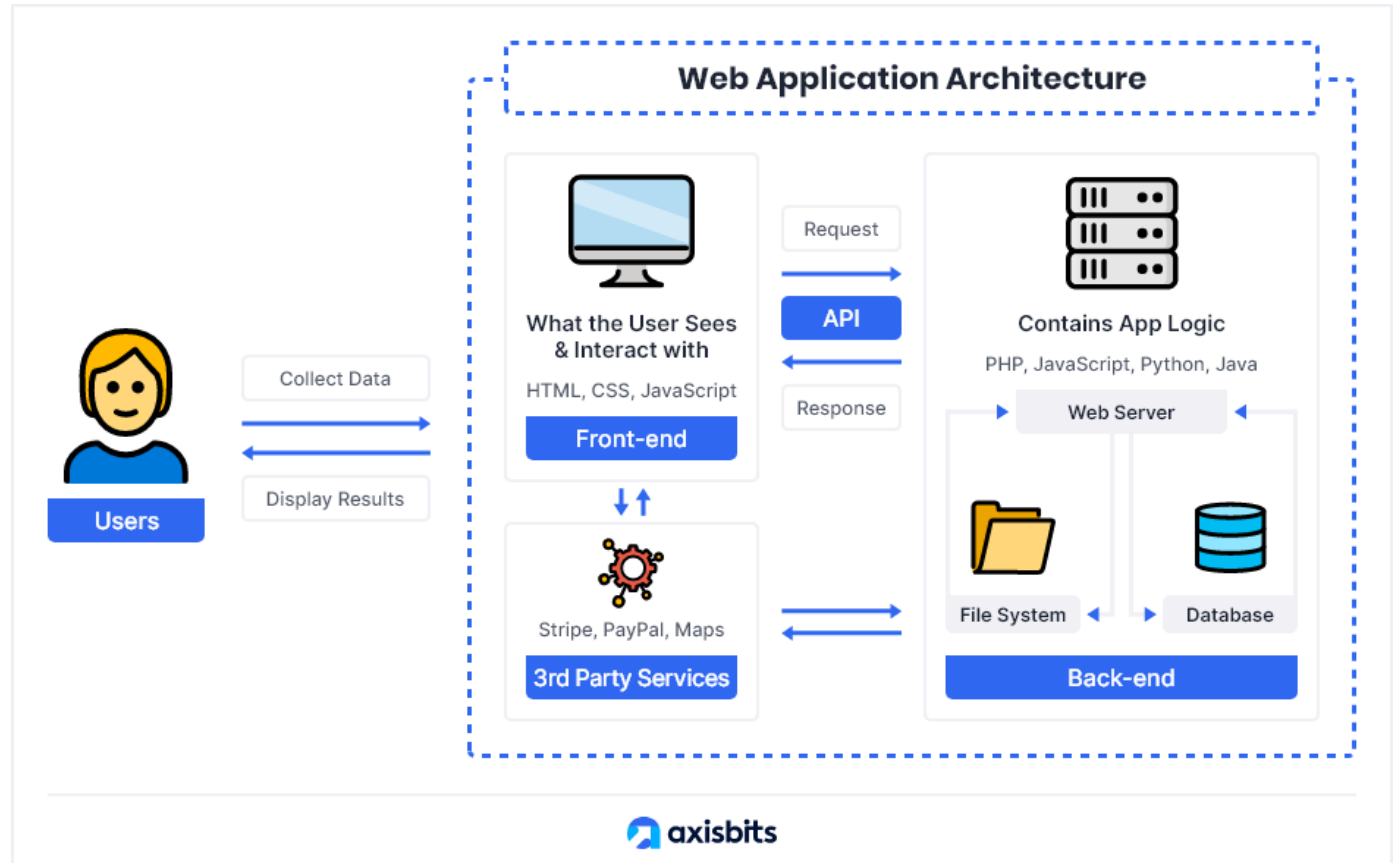
➤ What is software architecture?

- It is the overall structure of software system in terms of **components** and **connections**.
- It captures the components and the connections of the software.
- The components and connections helps to enforce design decisions that are made at the lower level.

□ Understanding basic terms *Contd...*

➤ What is software architecture?

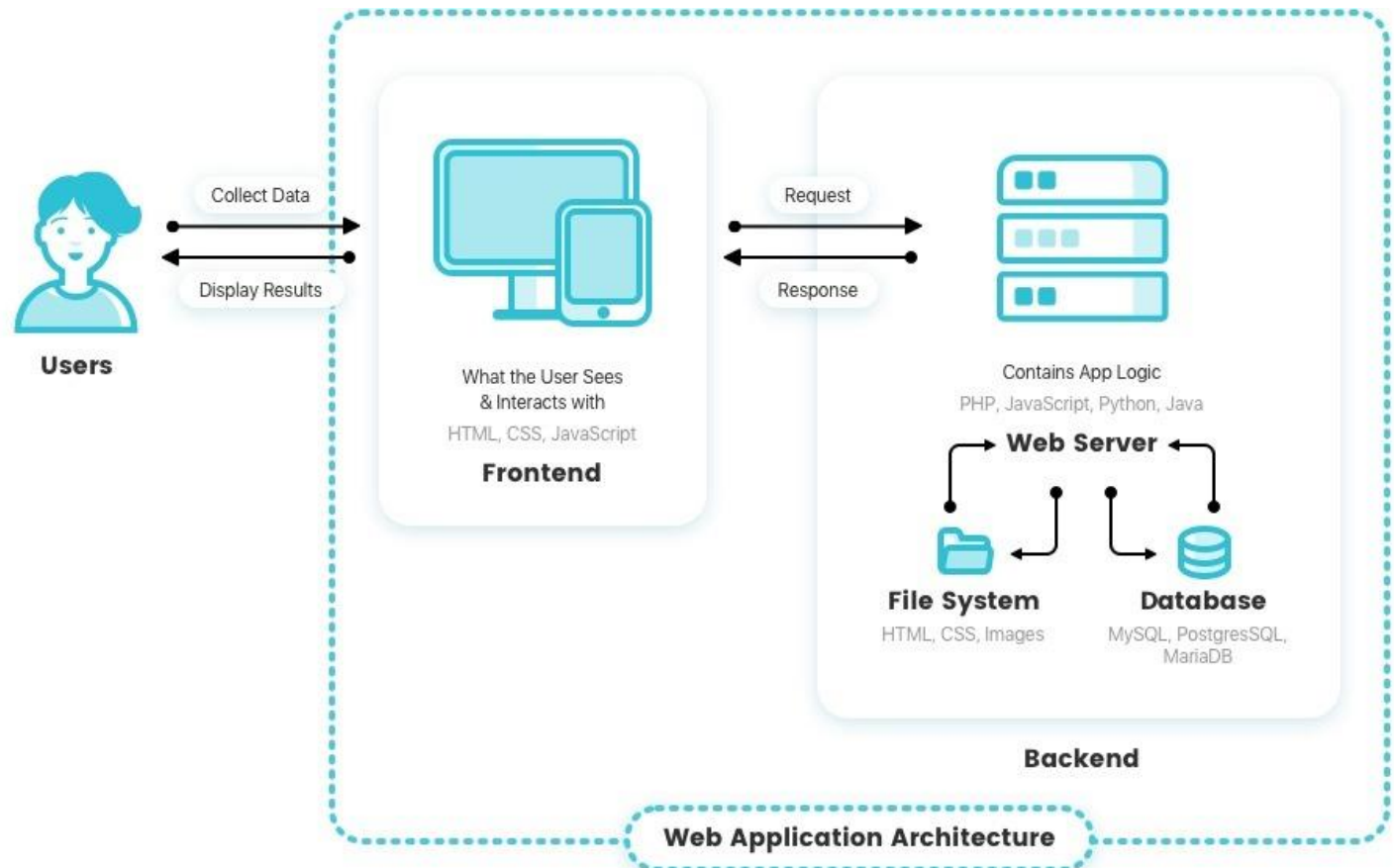
Example: 1



□ Understanding basic terms *Contd...*

➤ What is software architecture?

Example 2:



□ Understanding basic terms *Contd...*

➤ **What is software design?**

- Think of designing your software as you would a house.
- You start by drawing a rough sketch of the floor plan and layout of the rooms and floors.
- The drawing is your modeling language and the resulting blueprint will be a model of the final design.
- You will continue to modify the design until it arrive at a design that meets all your requirements.

□ Understanding basic terms Contd...

➤ **What is software design?**

- Activities involved in conceptualizing , framing , implementing, commissioning, and ultimately modifying complex systems.
- Activities following requirements specification and before programming.
- Describes the overall architecture and blueprint of software to be constructed.

□ Understanding basic terms *Contd...*

➤ **What is software design?**

- One of the benefit for designing your software using modeling language is that you discover problems early and fix them without refactoring your code.

□ Understanding basic terms Contd...

➤ **What is software design?**

- An important component of software design is software requirement analysis(SRA – list specification used)
- Two forms of software design

1. Process

- Sequence of steps that enables designer to describe all aspects of the software for building

2. Model

blueprint of software to be constructed (usually what is given to developers to write the code)

Understanding basic terms *Contd...*

➤ **What is a Software Process?**

- How do you know when to create/design software?
- Activities for designing, implementing, and testing a software system
- It is a lifecycle approach to development.
- It tells us the kind flow of activities in the software development. Is it waterfall, iterative etc.?

□ Understanding basic terms Contd...

➤ What is a Software Process?

- Representation of the order of activities of the process and the sequence in which they are performed.
- To provide guidance for controlling and coordinating the tasks to achieve the end product and objectives as effectively as possible.
- It helps us to organize our project.
- Lifecycle process controls the flow of activities.

□ Understanding basic terms Contd...

➤ What is a Software Design Method?

- This is usually documented in a software document plan (SDP) before starting a software development project.
- The combination of the *lifecycle process model* with *design/modeling guidelines* for clear guidance on which artifacts(object made by human) should be produced.

Why is modeling necessary

- Why don't we consider some basic requirements and then go straight to programming?
- Programming with the basic requirements is no longer feasible for most application.
- Software size is increasing exponentially.

Why is modeling necessary

- Most problems with software systems occur when different pieces have to interact.
- Still a poorly understood problem
- Problems are often discovered late with great cost.
- Often leads to performance issues too

□ The Need for Models

- A model is an abstraction, representing varying layers and views of complex information.
- Standard practice in nearly every engineering discipline.
 - Model helps us
 - ✓ Organize
 - ✓ Communicate
 - ✓ Reason
 - ✓ Analyze our system.

□ The Need for Models(1)

- We use models to develop large scale systems, organize them into manageable pieces and to communicate what we are building to our customers.
- With the models, we will reason and analyze the systems we are building before we begin writing a line of code.

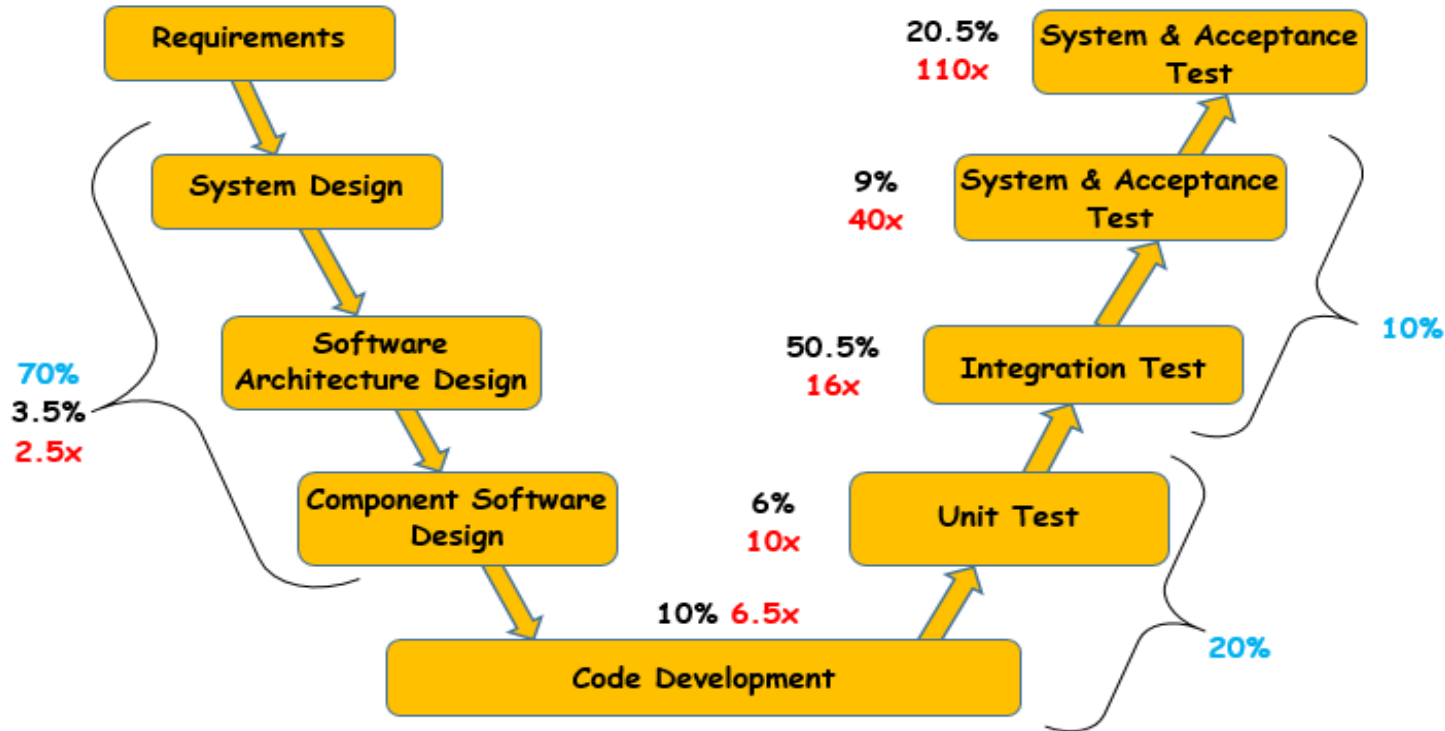
□ The Need for Models(2)

- We could analyze how the system will work by simulating the execution before writing the code.
- We can see the performance characteristics to get out of the system.
- It is easier and cheaper to fix errors in software design level models than to fix things when coding.

The Need for Models(3)

- Software models are ways of expressing a software design while software modeling expressing the entire software design including interfaces, interactions with other software, and all the software methods.

□ Impact of software fault



Key:

- Where faults are introduced
- Where faults are found
- Estimated cost factor for fault removal

Impact of software fault

- Basically the model is the primary software engineering artifact at this point.
- Model-based software engineering approaches provide a consistent, unified model supporting analysis from the earliest stages of the software engineering lifecycle.

Impact of Software fault

- Basically the model is the primary software engineering artifact at this point.
- Model-based software engineering approaches provide a consistent, unified model supporting analysis from the earliest stages of the software engineering lifecycle.

QUESTIONS



Class Evaluation

- **Groups (7 members)**
- **Creating of groups**
- **Use of Trello and Slack (creation and Usage)**
- **Attendance recording (daily and weekly)**
- **UML software : StarUML**