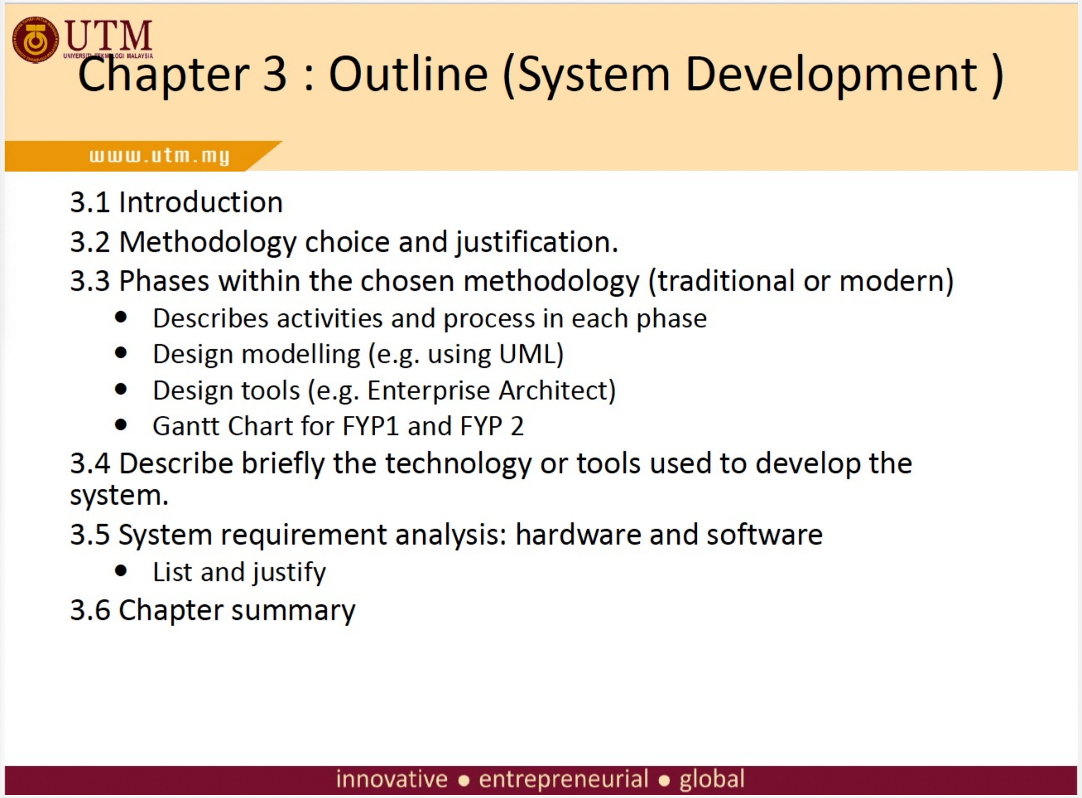
Chapter 3
System Development Methodology



Chapter 3 : Outline (System Development)

www.utm.my

3.1 Introduction

3.2 Methodology choice and justification.

3.3 Phases within the chosen methodology (traditional or modern)

- Describes activities and process in each phase
- Design modelling (e.g. using UML)
- Design tools (e.g. Enterprise Architect)
- Gantt Chart for FYP1 and FYP 2

3.4 Describe briefly the technology or tools used to develop the system.

3.5 System requirement analysis: hardware and software

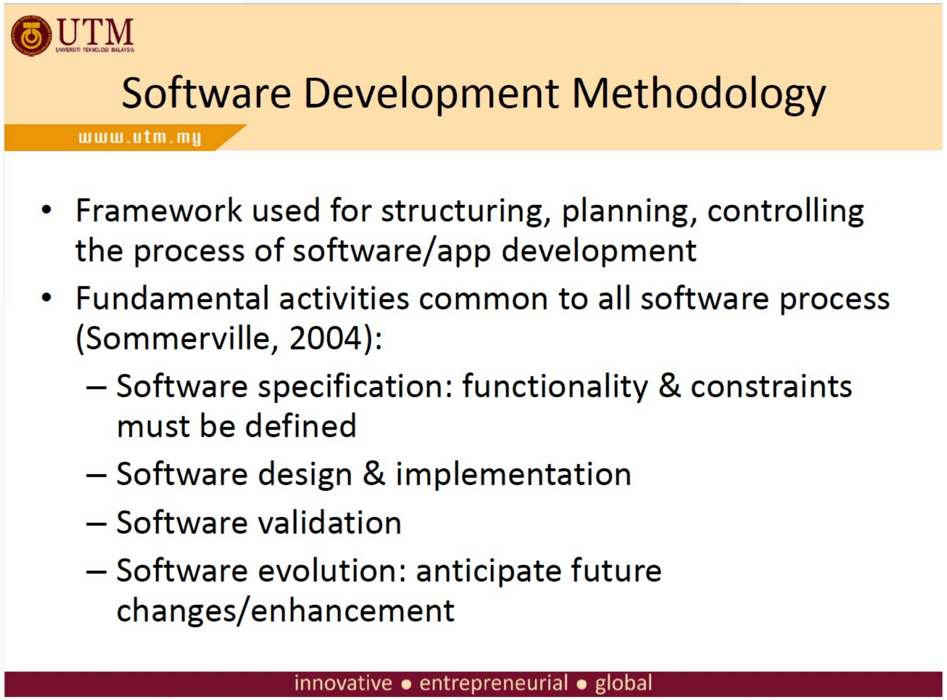
- List and justify

3.6 Chapter summary

innovative • entrepreneurial • global

Will re-visit the Chapter's structure at the end

What is… ?



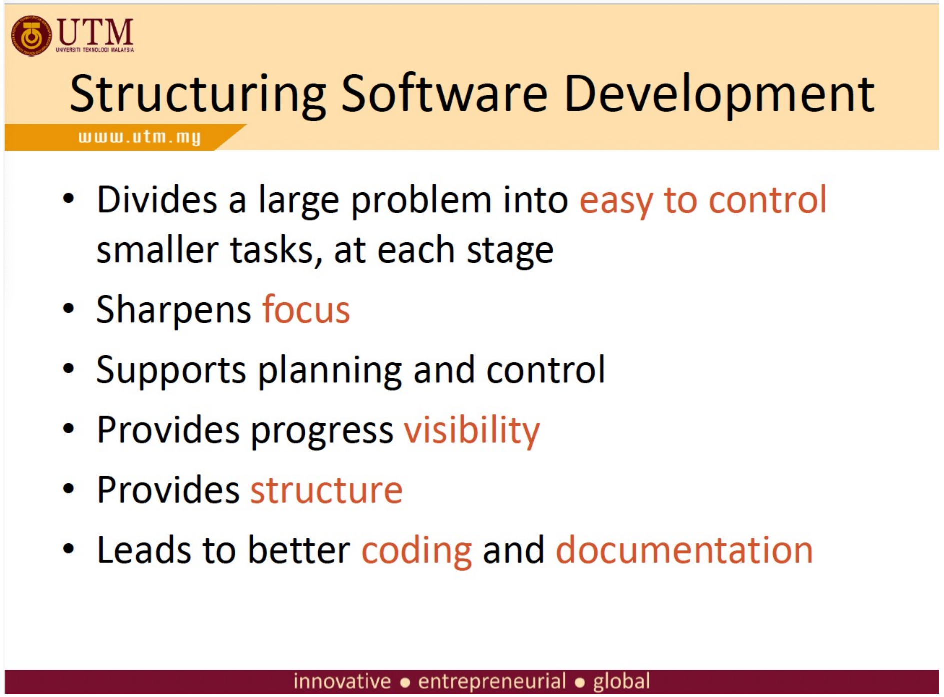
UTM UNIVERSITI TEKNOLOGI MALAYSIA

Software Development Methodology

www.utm.my

- Framework used for structuring, planning, controlling the process of software/app development
- Fundamental activities common to all software process (Sommerville, 2004):
  - Software specification: functionality & constraints must be defined
  - Software design & implementation
  - Software validation
  - Software evolution: anticipate future changes/enhancement

innovative ● entrepreneurial ● global



UTM UNIVERSITI TEKNOLOGI MALAYSIA

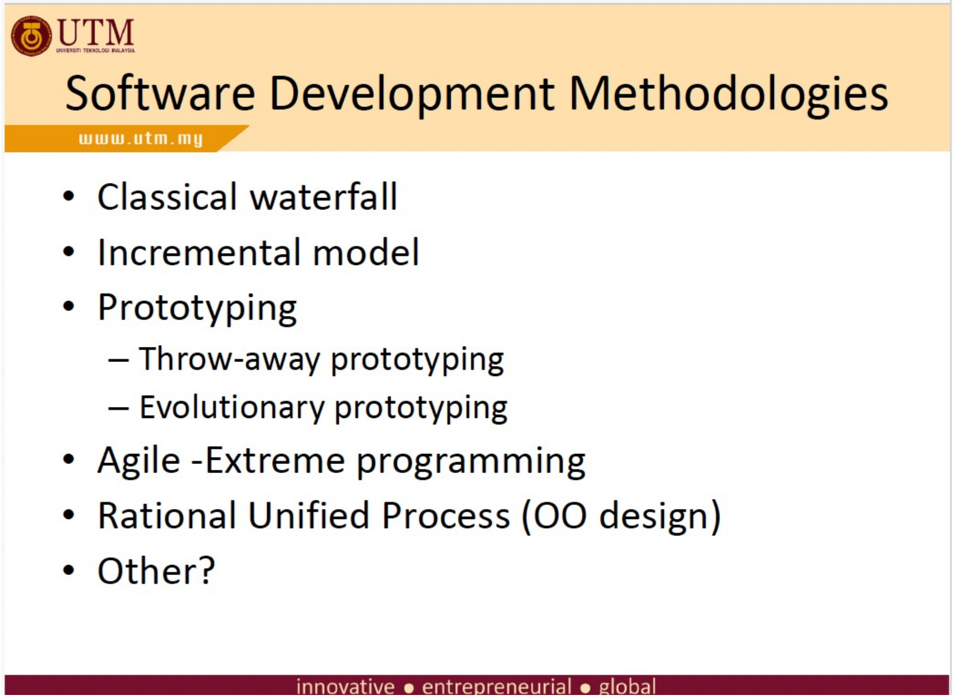
Structuring Software Development

www.utm.my

- Divides a large problem into easy to control smaller tasks, at each stage
- Sharpens focus
- Supports planning and control
- Provides progress visibility
- Provides structure
- Leads to better coding and documentation

innovative ● entrepreneurial ● global

What we already know?



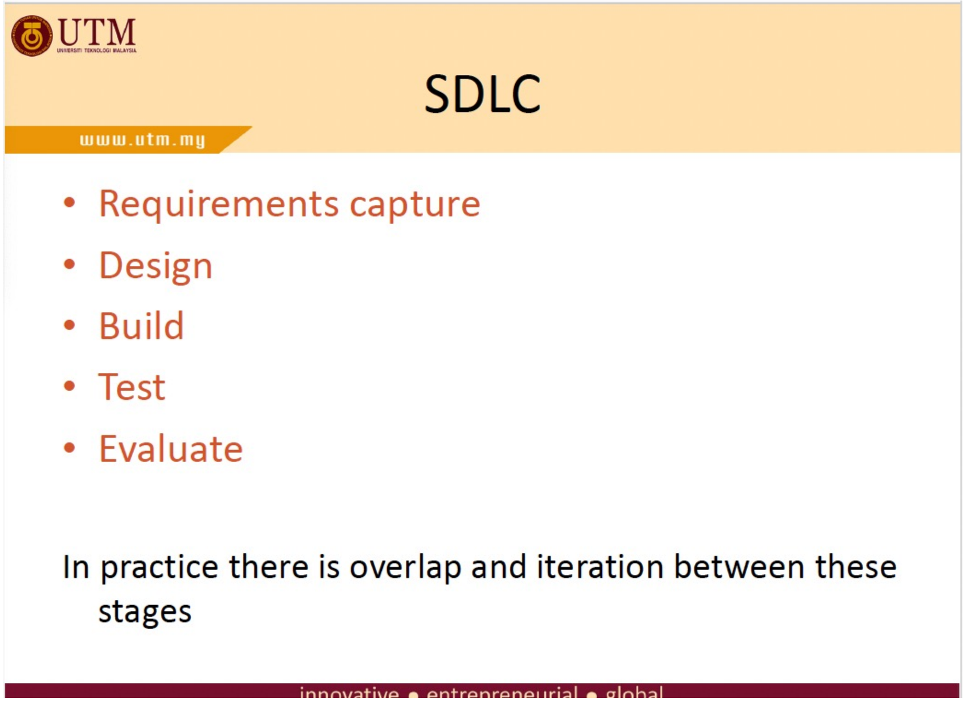
UTM

www.utm.my

Software Development Methodologies

- Classical waterfall
- Incremental model
- Prototyping
  - Throw-away prototyping
  - Evolutionary prototyping
- Agile -Extreme programming
- Rational Unified Process (OO design)
- Other?

innovative ● entrepreneurial ● global



UTM

www.utm.my

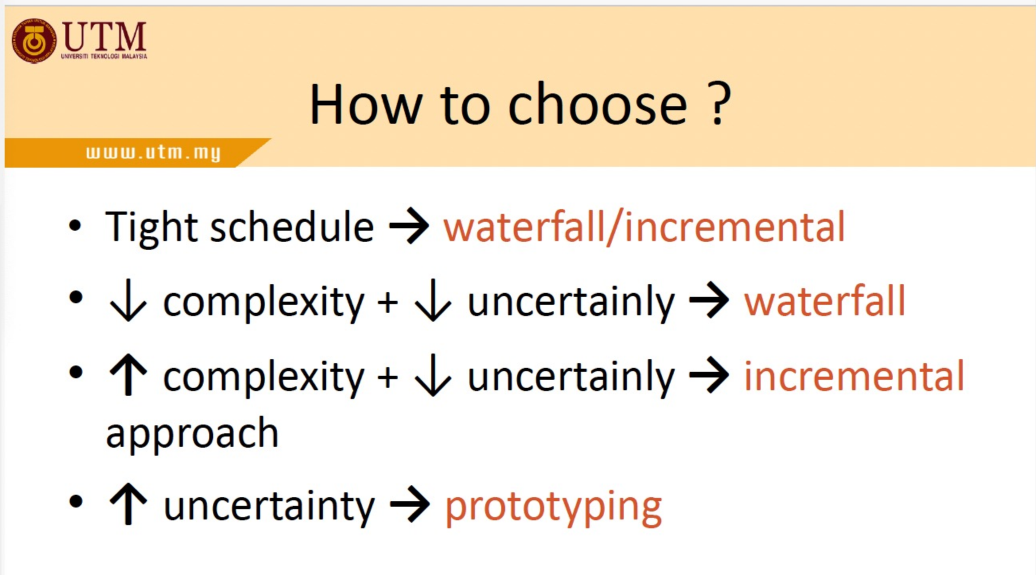
SDLC

- Requirements capture
- Design
- Build
- Test
- Evaluate

In practice there is overlap and iteration between these stages

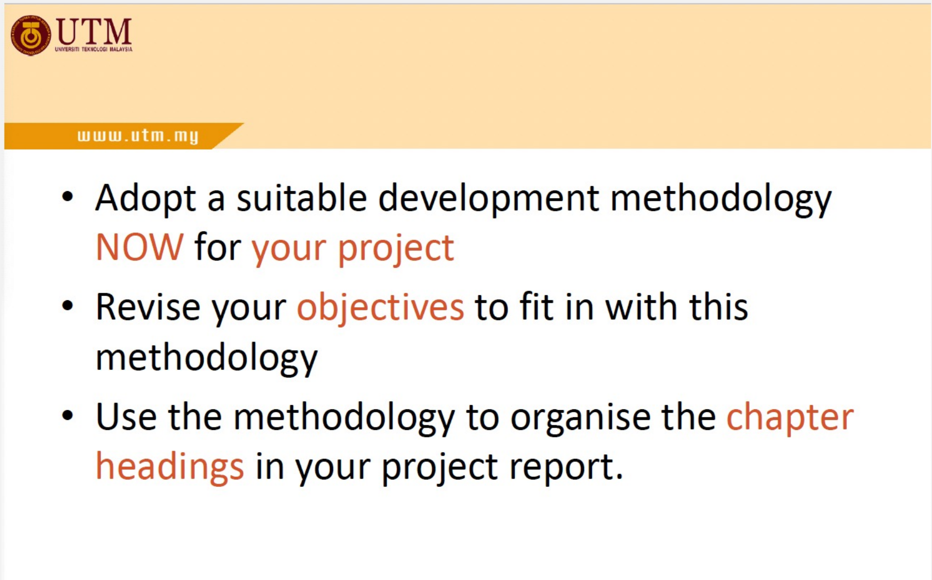
innovative ● entrepreneurial ● global

How this affects the project?



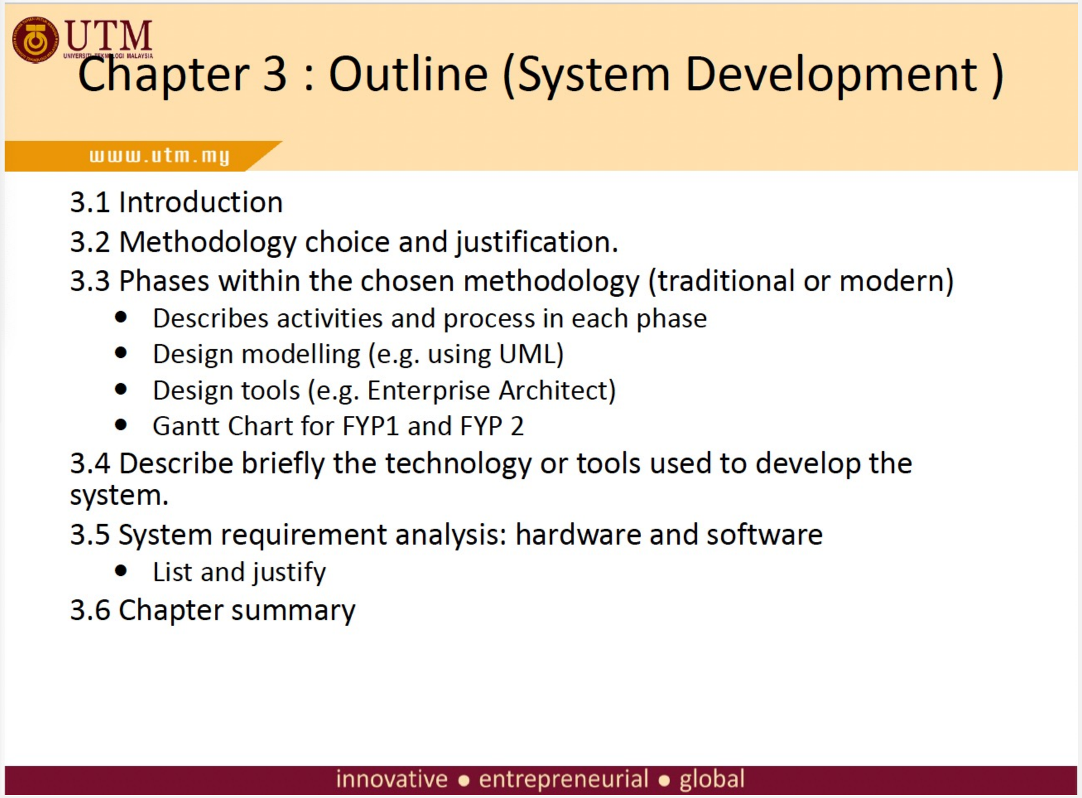
How to choose ?

- Tight schedule → waterfall/incremental
- ↓ complexity + ↓ uncertainly → waterfall
- ↑ complexity + ↓ uncertainly → incremental approach
- ↑ uncertainty → prototyping



- Adopt a suitable development methodology NOW for your project
- Revise your objectives to fit in with this methodology
- Use the methodology to organise the chapter headings in your project report.

Let’s look at few samples



Chapter 3 : Outline (System Development)

www.utm.my

3.1 Introduction

3.2 Methodology choice and justification.

3.3 Phases within the chosen methodology (traditional or modern)

- Describes activities and process in each phase
- Design modelling (e.g. using UML)
- Design tools (e.g. Enterprise Architect)
- Gantt Chart for FYP1 and FYP 2

3.4 Describe briefly the technology or tools used to develop the system.

3.5 System requirement analysis: hardware and software

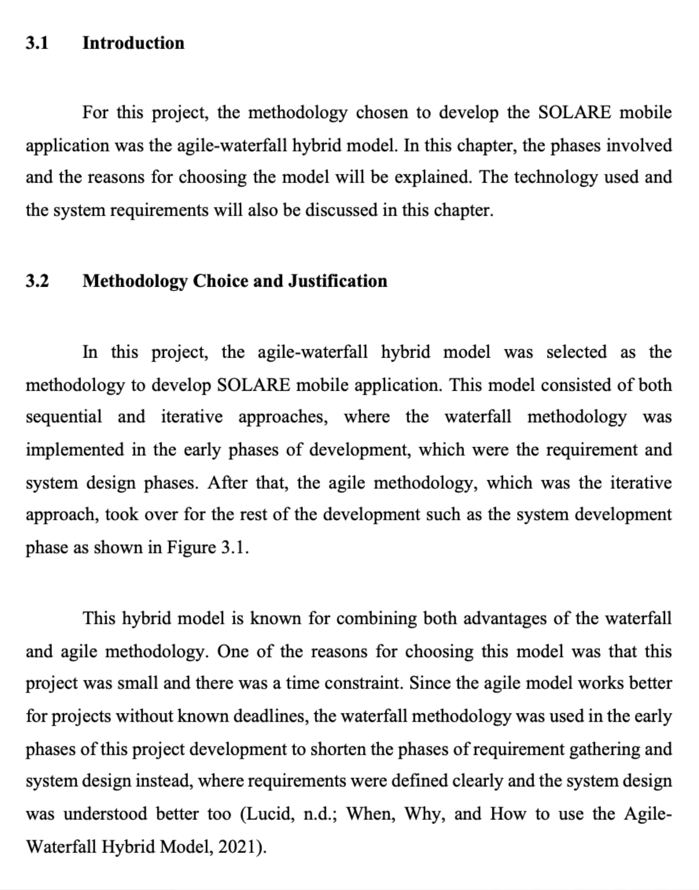
- List and justify

3.6 Chapter summary

innovative ● entrepreneurial ● global

Waterfall-agile

Section 3.1-3.3



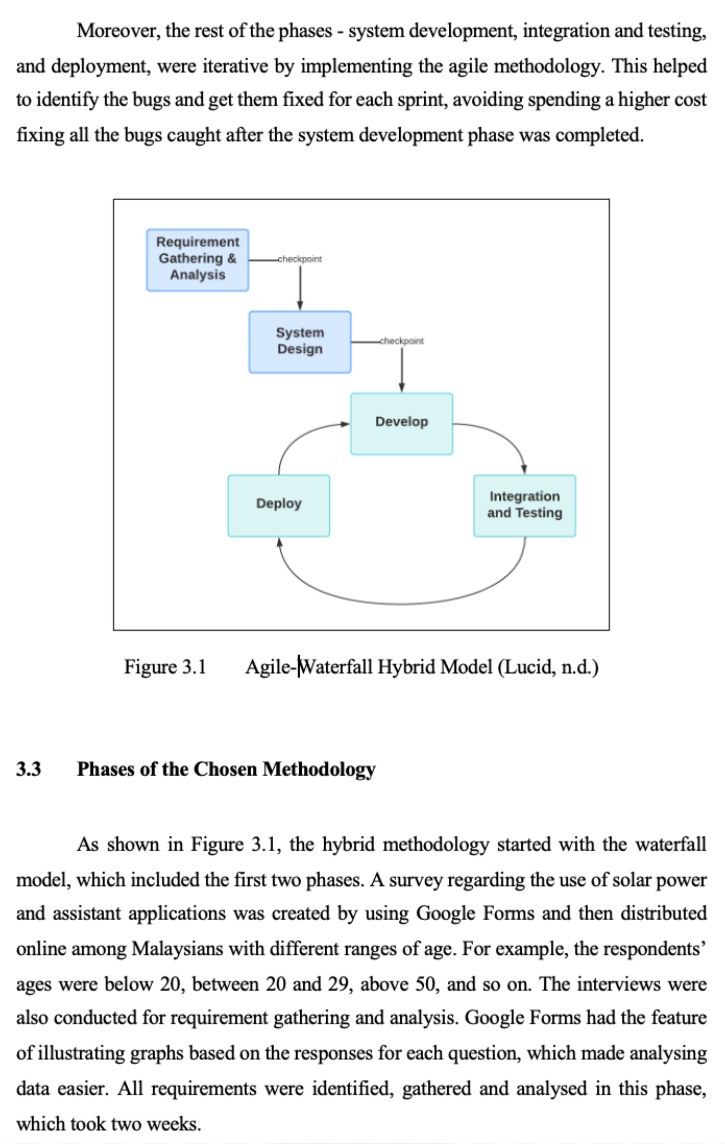
3.1 Introduction

For this project, the methodology chosen to develop the SOLARE mobile application was the agile-waterfall hybrid model. In this chapter, the phases involved and the reasons for choosing the model will be explained. The technology used and the system requirements will also be discussed in this chapter.

3.2 Methodology Choice and Justification

In this project, the agile-waterfall hybrid model was selected as the methodology to develop SOLARE mobile application. This model consisted of both sequential and iterative approaches, where the waterfall methodology was implemented in the early phases of development, which were the requirement and system design phases. After that, the agile methodology, which was the iterative approach, took over for the rest of the development such as the system development phase as shown in Figure 3.1.

This hybrid model is known for combining both advantages of the waterfall and agile methodology. One of the reasons for choosing this model was that this project was small and there was a time constraint. Since the agile model works better for projects without known deadlines, the waterfall methodology was used in the early phases of this project development to shorten the phases of requirement gathering and system design instead, where requirements were defined clearly and the system design was understood better too (Lucid, n.d.; When, Why, and How to use the Agile-Waterfall Hybrid Model, 2021).



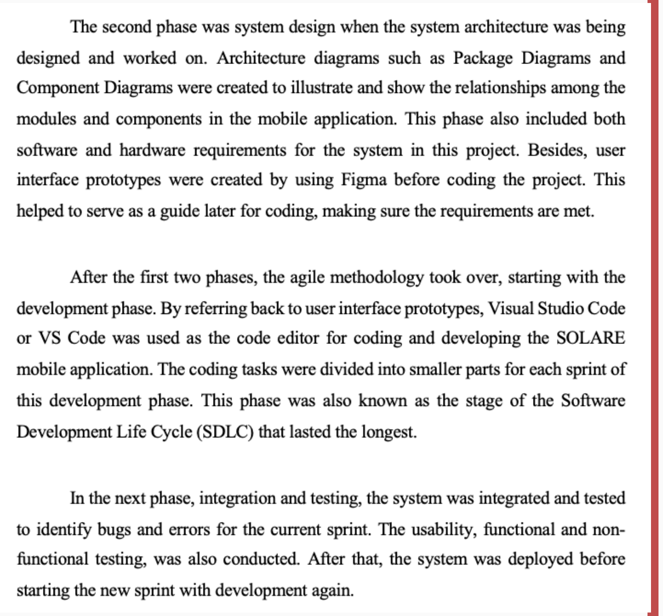
Moreover, the rest of the phases - system development, integration and testing, and deployment, were iterative by implementing the agile methodology. This helped to identify the bugs and get them fixed for each sprint, avoiding spending a higher cost fixing all the bugs caught after the system development phase was completed.

Requirement Gathering & Analysis
checkpoint
System Design
checkpoint
Develop
Integration and Testing
Deploy

Figure 3.1 Agile-Waterfall Hybrid Model (Lucid, n.d.)

3.3 Phases of the Chosen Methodology

As shown in Figure 3.1, the hybrid methodology started with the waterfall model, which included the first two phases. A survey regarding the use of solar power and assistant applications was created by using Google Forms and then distributed online among Malaysians with different ranges of age. For example, the respondents' ages were below 20, between 20 and 29, above 50, and so on. The interviews were also conducted for requirement gathering and analysis. Google Forms had the feature of illustrating graphs based on the responses for each question, which made analysing data easier. All requirements were identified, gathered and analysed in this phase, which took two weeks.



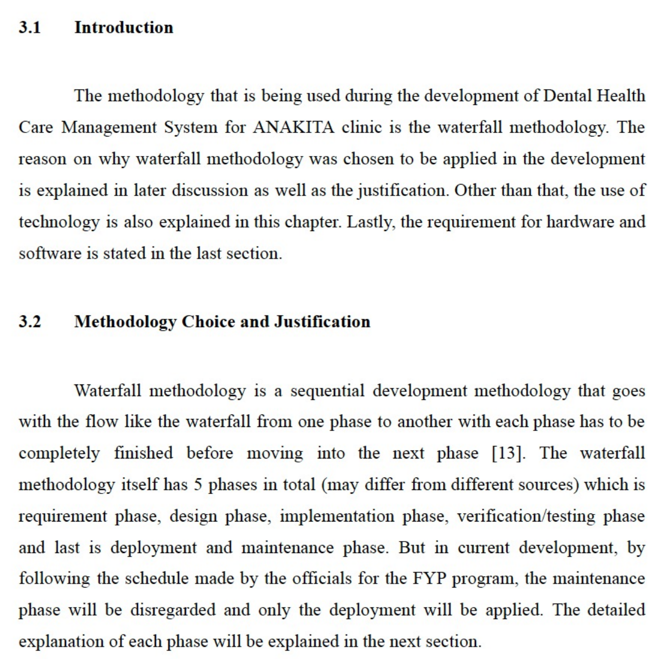
The second phase was system design when the system architecture was being designed and worked on. Architecture diagrams such as Package Diagrams and Component Diagrams were created to illustrate and show the relationships among the modules and components in the mobile application. This phase also included both software and hardware requirements for the system in this project. Besides, user interface prototypes were created by using Figma before coding the project. This helped to serve as a guide later for coding, making sure the requirements are met.

After the first two phases, the agile methodology took over, starting with the development phase. By referring back to user interface prototypes, Visual Studio Code or VS Code was used as the code editor for coding and developing the SOLARE mobile application. The coding tasks were divided into smaller parts for each sprint of this development phase. This phase was also known as the stage of the Software Development Life Cycle (SDLC) that lasted the longest.

In the next phase, integration and testing, the system was integrated and tested to identify bugs and errors for the current sprint. The usability, functional and non-functional testing, was also conducted. After that, the system was deployed before starting the new sprint with development again.

Waterfall

Section 3.1-3.3



3.1 Introduction

The methodology that is being used during the development of Dental Health Care Management System for ANAKITA clinic is the waterfall methodology. The reason on why waterfall methodology was chosen to be applied in the development is explained in later discussion as well as the justification. Other than that, the use of technology is also explained in this chapter. Lastly, the requirement for hardware and software is stated in the last section.

3.2 Methodology Choice and Justification

Waterfall methodology is a sequential development methodology that goes with the flow like the waterfall from one phase to another with each phase has to be completely finished before moving into the next phase [13]. The waterfall methodology itself has 5 phases in total (may differ from different sources) which is requirement phase, design phase, implementation phase, verification/testing phase and last is deployment and maintenance phase. But in current development, by following the schedule made by the officials for the FYP program, the maintenance phase will be disregarded and only the deployment will be applied. The detailed explanation of each phase will be explained in the next section.

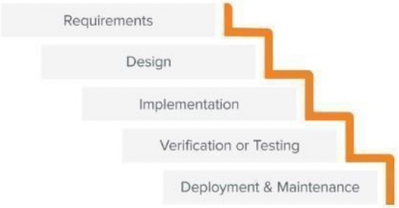
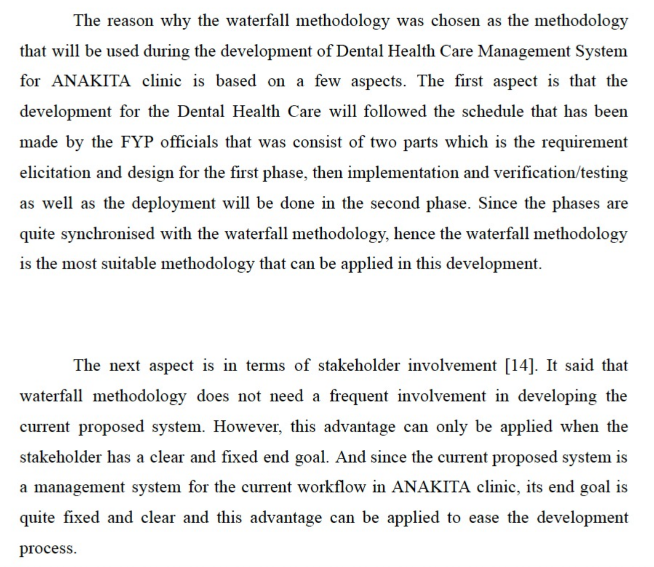
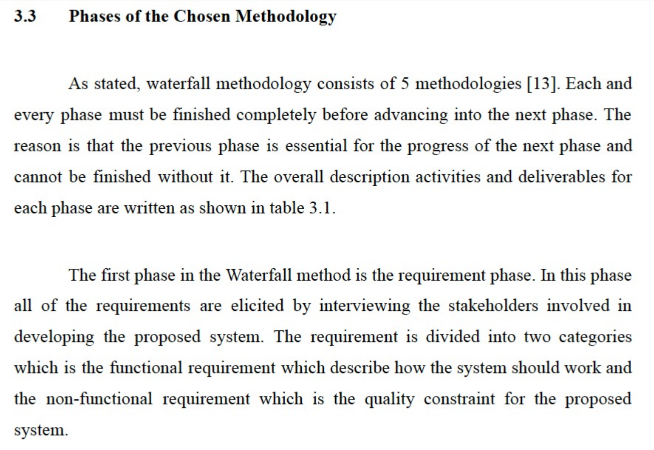


Figure 3.1: Waterfall method's phases



The reason why the waterfall methodology was chosen as the methodology that will be used during the development of Dental Health Care Management System for ANAKITA clinic is based on a few aspects. The first aspect is that the development for the Dental Health Care will followed the schedule that has been made by the FYP officials that was consist of two parts which is the requirement elicitation and design for the first phase, then implementation and verification/testing as well as the deployment will be done in the second phase. Since the phases are quite synchronised with the waterfall methodology, hence the waterfall methodology is the most suitable methodology that can be applied in this development.

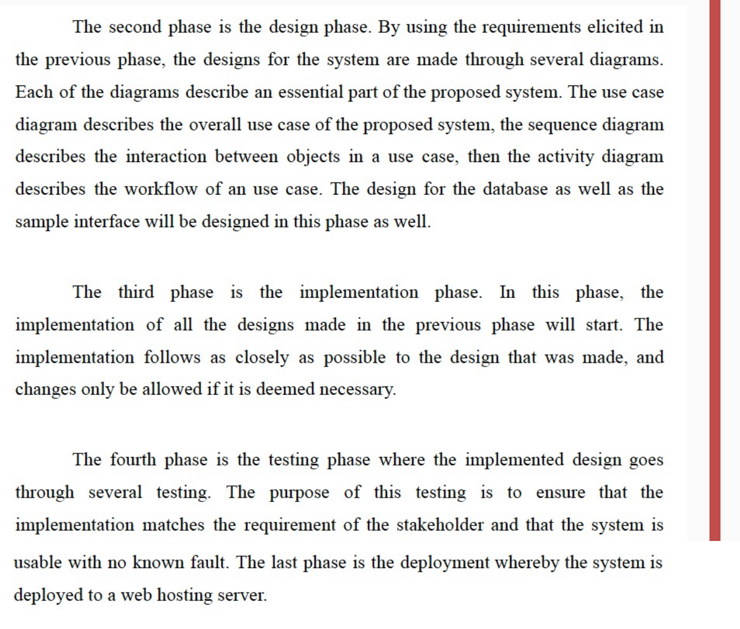
The next aspect is in terms of stakeholder involvement [14]. It said that waterfall methodology does not need a frequent involvement in developing the current proposed system. However, this advantage can only be applied when the stakeholder has a clear and fixed end goal. And since the current proposed system is a management system for the current workflow in ANAKITA clinic, its end goal is quite fixed and clear and this advantage can be applied to ease the development process.



3.3 Phases of the Chosen Methodology

As stated, waterfall methodology consists of 5 methodologies [13]. Each and every phase must be finished completely before advancing into the next phase. The reason is that the previous phase is essential for the progress of the next phase and cannot be finished without it. The overall description activities and deliverables for each phase are written as shown in table 3.1.

The first phase in the Waterfall method is the requirement phase. In this phase all of the requirements are elicited by interviewing the stakeholders involved in developing the proposed system. The requirement is divided into two categories which is the functional requirement which describe how the system should work and the non-functional requirement which is the quality constraint for the proposed system.

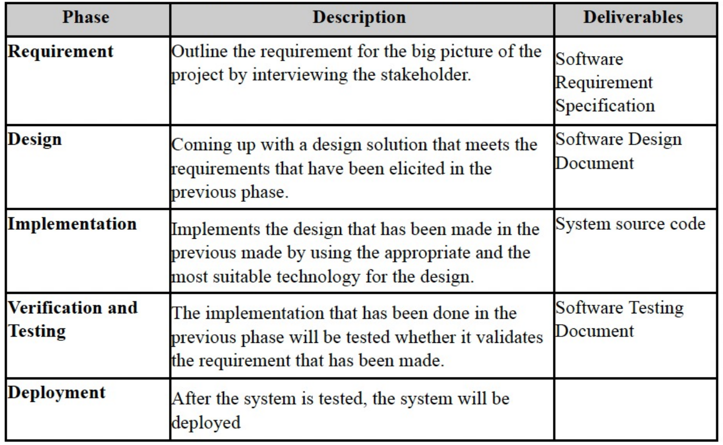


The second phase is the design phase. By using the requirements elicited in the previous phase, the designs for the system are made through several diagrams. Each of the diagrams describe an essential part of the proposed system. The use case diagram describes the overall use case of the proposed system, the sequence diagram describes the interaction between objects in a use case, then the activity diagram describes the workflow of an use case. The design for the database as well as the sample interface will be designed in this phase as well.

The third phase is the implementation phase. In this phase, the implementation of all the designs made in the previous phase will start. The implementation follows as closely as possible to the design that was made, and changes only be allowed if it is deemed necessary.

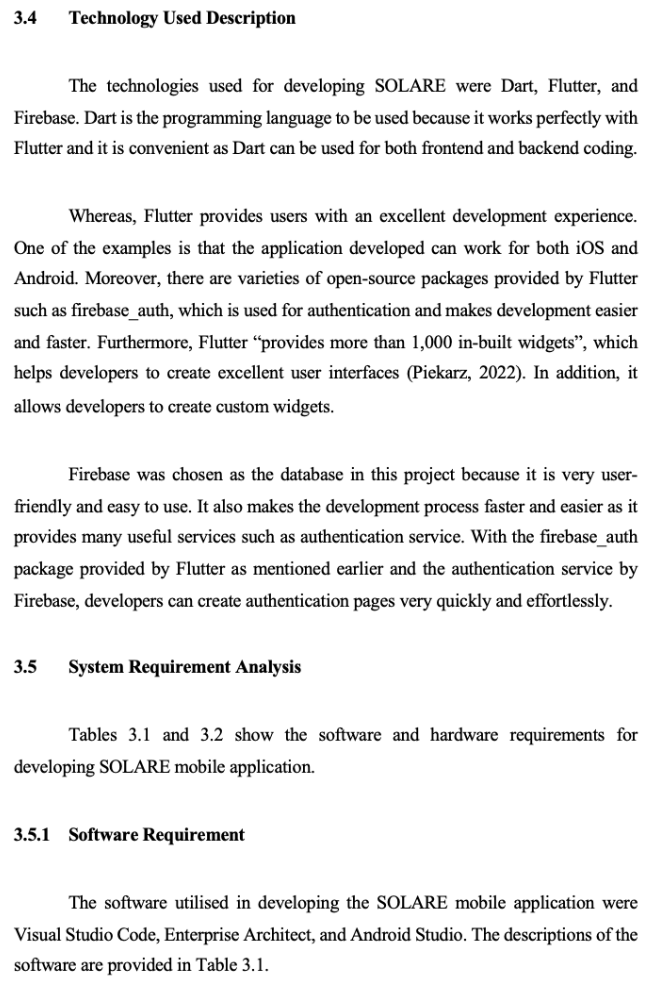
The fourth phase is the testing phase where the implemented design goes through several testing. The purpose of this testing is to ensure that the implementation matches the requirement of the stakeholder and that the system is usable with no known fault. The last phase is the deployment whereby the system is deployed to a web hosting server.

Table 3.1: Waterfall method phase explanation summary



Phase	Description	Deliverables
Requirement	Outline the requirement for the big picture of the project by interviewing the stakeholder.	Software Requirement Specification
Design	Coming up with a design solution that meets the requirements that have been elicited in the previous phase.	Software Design Document
Implementation	Implements the design that has been made in the previous made by using the appropriate and the most suitable technology for the design.	System source code
Verification and Testing	The implementation that has been done in the previous phase will be tested whether it validates the requirement that has been made.	Software Testing Document
Deployment	After the system is tested, the system will be deployed	

Section 3.4-3.6



3.4 Technology Used Description

The technologies used for developing SOLARE were Dart, Flutter, and Firebase. Dart is the programming language to be used because it works perfectly with Flutter and it is convenient as Dart can be used for both frontend and backend coding.

Whereas, Flutter provides users with an excellent development experience. One of the examples is that the application developed can work for both iOS and Android. Moreover, there are varieties of open-source packages provided by Flutter such as firebase_auth, which is used for authentication and makes development easier and faster. Furthermore, Flutter "provides more than 1,000 in-built widgets", which helps developers to create excellent user interfaces (Piekarz, 2022). In addition, it allows developers to create custom widgets.

Firebase was chosen as the database in this project because it is very user-friendly and easy to use. It also makes the development process faster and easier as it provides many useful services such as authentication service. With the firebase_auth package provided by Flutter as mentioned earlier and the authentication service by Firebase, developers can create authentication pages very quickly and effortlessly.

3.5 System Requirement Analysis

Tables 3.1 and 3.2 show the software and hardware requirements for developing SOLARE mobile application.

3.5.1 Software Requirement

The software utilised in developing the SOLARE mobile application were Visual Studio Code, Enterprise Architect, and Android Studio. The descriptions of the software are provided in Table 3.1.

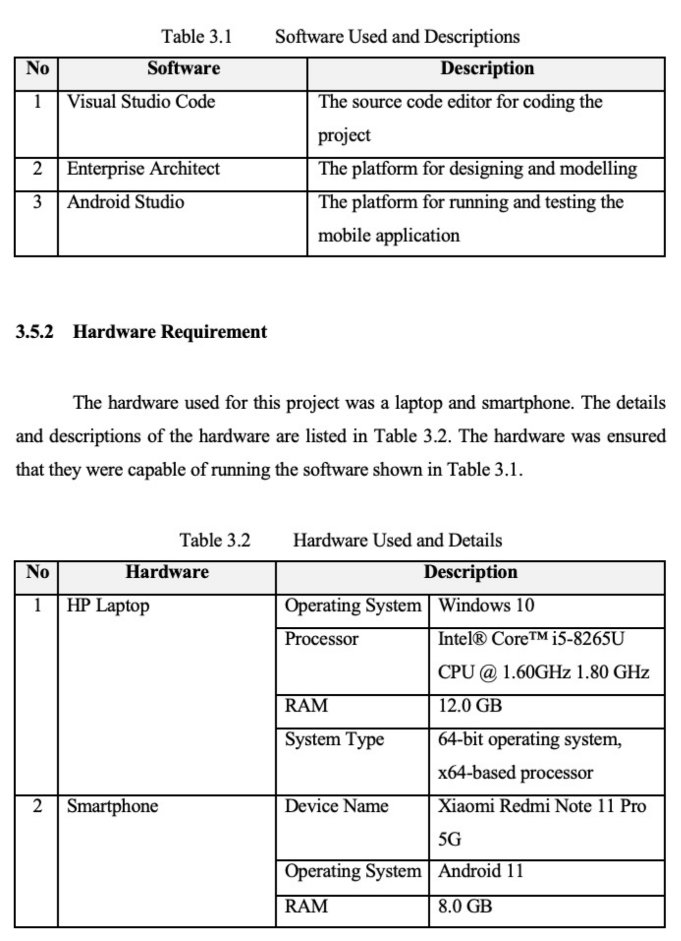


Table 3.1 Software Used and Descriptions

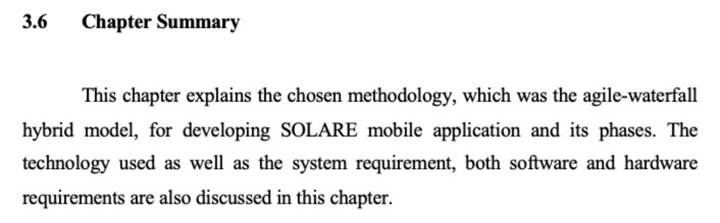
No	Software	Description
1	Visual Studio Code	The source code editor for coding the project
2	Enterprise Architect	The platform for designing and modelling
3	Android Studio	The platform for running and testing the mobile application

3.5.2 Hardware Requirement

The hardware used for this project was a laptop and smartphone. The details and descriptions of the hardware are listed in Table 3.2. The hardware was ensured that they were capable of running the software shown in Table 3.1.

Table 3.2 Hardware Used and Details

No	Hardware	Description	
1	HP Laptop	Operating System	Windows 10
		Processor	Intel® Core™ i5-8265U CPU @ 1.60GHz 1.80 GHz
		RAM	12.0 GB
		System Type	64-bit operating system, x64-based processor
2	Smartphone	Device Name	Xiaomi Redmi Note 11 Pro 5G
		Operating System	Android 11
		RAM	8.0 GB



3.6 Chapter Summary

This chapter explains the chosen methodology, which was the agile-waterfall hybrid model, for developing SOLARE mobile application and its phases. The technology used as well as the system requirement, both software and hardware requirements are also discussed in this chapter.

Students’ reports:

Lai Ting Ying, July 2023, SOLARE: Solar Power Assistant Application

Muhammad Rafiy Athalla, July 2023, Dental Health Care Management System for Anakita Clinic