

**DIPLOMA IN INFORMATION TECHNOLOGY**

**SOFTWARE ENGINEERING (CGE2C09)**

**AY 2017/2018 APR SEMESTER**

**ASSIGNMENT 1**

**Tutor: Mr Mel Goh**

**Class: P02**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1500169F** | **Camille Pamela Anne S. Sioson** |
| **1502015I** | **Dixon Low Yi Sheng** |
| **1506381A** | **Raeburn Timothy Nigel** |
| **1502527C** | **Muhammad Nur Afnan Bin Abdul Rahim** |

**Appendix B: Declaration of Work of Originality**

**Software Engineering (CGE2C09)**

**AY 2017/2018 Apr Semester**

**Practical Class**:  P02

**Submitted by**: <1500169F> <Camille Pamela Anne S. Sioson>

<1502015I> <Dixon Low Yi Sheng >  
 <1502527C><Muhammad Nur Afnan Bin Abdul Rahim>

<1506381A> <Raeburn Timothy Nigel>

**Date:**  9/5/2017

**“By submitting this work, I am / we are declaring that I am / we are the originator(s) of this work and that all other original sources used in this work has been appropriately acknowledged.**

**I / We understand that plagiarism is the act of taking and using the whole or any part of another person’s work and presenting it as my/ our own without proper acknowledgement.**

**I / We also understand that plagiarism is an academic offence and that disciplinary action will be taken for plagiarism.”**

**Name and Signature of student: ……………………………………**

**Name and Signature of student: ……………………………………**

**Name and Signature of student: ……………………………………**

**Name and Signature of student: ……………………………………**

Contents

[Software Development Life Cycle (SDLC) 1](#_Toc482069079)

[Camille Pamela Anne S. Sioson (1500169F) 1](#_Toc482069080)

[Raeburn Timothy Nigel (1506381A) 6](#_Toc482069081)

[Muhammad Nur Afnan Bin Abdul Rahim (1502527C) 9](#_Toc482069082)

[Dixon Low Yi Sheng (1502015I) 13](#_Toc482069083)

# **Software Development Life Cycle (SDLC)**

## **Camille Pamela Anne S. Sioson (1500169F)**

**Software Development Life Cycle (SDLC)**

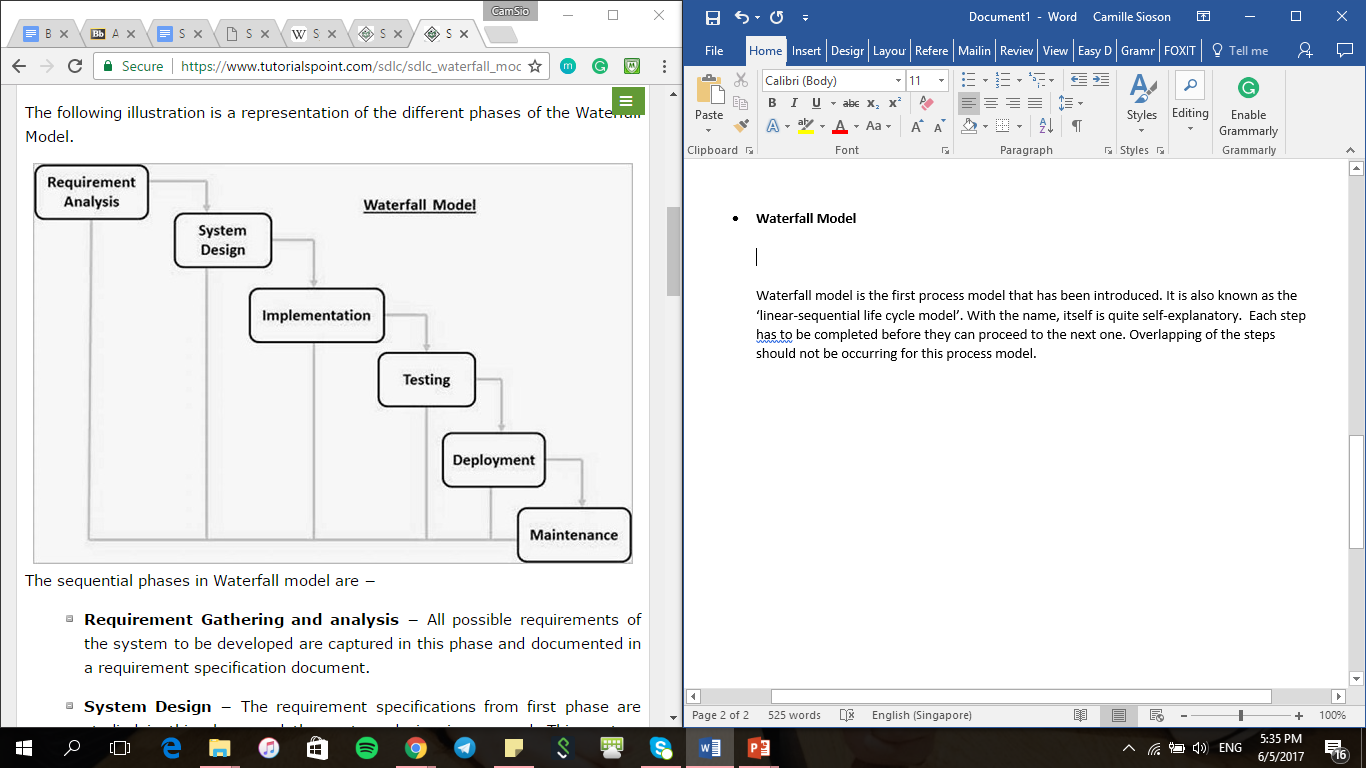
There are 5 phases involved in Software Development Life Cycle (SDLC). The 5 phases are:

* **Requirements Engineering**it is also known as the specification. This is the phase whereby the client/customer states what they want the software to do or have. however, at this stage it might not be clear as of what they want due to some circumstances such as not specifying the requirements precisely. This then leads to giving incomplete requirements and misunderstanding user goals.
* **Analysis and Design**Analysis uses both text and diagrams to demonstrate or rather, show the requirements. Design is to produce a representation of an entity such as having Use-Case diagram, Sequence Diagram, Database design etc. This stage does have its pitfalls as well. There can be many deigns done, so choosing which is better and which would fit best may be a problem faced. Speaking of designs, having a lot of designs as well, it may lead to compromising it due to issues like lack of time or insufficient budget to fulfil such designs.
* **Implementation (whereby coding is involved)**This is the phase where the chosen design or rather, the ideal detailed designs is converted into steps written in the programming language. Having not to meet the deadline may lead them to compromising the ideas, which may or may not be a bad idea and during the implementation period, there can be confliction choices such as cost versus time etc.
* **Testing**Once everything has been implemented, testing is required to be done. This is so to make sure that the software is reliable and meets the users’ needs as what they have addressed during the first phase which was requirements engineering.
* **Deployment**This is the final step in SDLC where the application/software system that has been done will be distributed to the selected customers prior to official release. After which, it will be delivered to the customer. And this may require training to be able to teach and guide especially if first-time users.

SDLC is used by system engineers and system developers to be able to deliver high-quality systems. It is a framework which defines the tasks that is needed to be fulfilled in the respective phases. There are different types of SDLC models defined and designed. Each model is unique of its own to its type to make sure of success in the process of software development. These are the more popular and important SDLC models and is covered in our lecture:

* Waterfall Model
* Prototyping
* Unified process
* Agile methods

I will be talking through and explaining **Waterfall Model, Prototyping** and **Agile methods** more in depth.

**Waterfall Model**

Waterfall model is the first process model that has been introduced. It is also known as the ‘linear-sequential life cycle model’. With the name, itself is quite self-explanatory. Each step must be completed before they can proceed to the next one. Overlapping of the steps should not be occurring for this process model. The illustration above represents the flow of the steps of the Waterfall model. The sequence is as follows:

1. **Requirement gathering and analysis**

This step ensures that all the requirements needed for the software is gathered, captured and documented in a document to avoid any points being missed out or any requirements that are critical for the system that has to be developed.

1. **System Design**

The specifications that was gathered and documented on the first step will then needed to be studied in this step. After which, the system design is prepared. The system design helps to figure out which hardware is needed and other system requirements as well. This helps in further enhancing the overall system architecture.

1. **Implementation**

The inputs that was gathered from the system design is then used. The system is first developed in small programs called units. These units will then be integrated in the following step. Each unit is developed and tested for its functions for the system. This process is called unit testing.

1. **Integration and Testing**

The units that were developed in the previous step will be integrated into a system, assuming it has passed the unit testing. After which, the system will be tested for any glitches, faults or/and failures.

1. **Deployment of system**

Once that all testing is done; both functional and non-functional, the product is deployed in the customer environment or/and released to the market.

1. **Maintenance**

Once that the system is deployed in the customer environment or to the market, there are cases where there will be some technical issues that will be faced by the clients. This is where maintenance comes in. to fix the issues that are raised by the clients, there will be patches that will be released. These patches are to resolve the issued. Also, in the technology today, it is fast-paced. Hence, having better versions to be released is also a part of maintenance. Maintenance is carried out to make these changes in the customer environment.

Every software is different; hence waterfall model is not suitable for all. The waterfall model is ideal only when these factors are present:

* Requirements are well-discussed and well-documented
* Definition of the product is stable
* The technology used is not dynamic and is well-understood
* The project is short (in the sense of the workload, or rather the duration of doing the project)

**Prototyping**

A prototype is a scaled down version of the actual system. With that being said, its functionalities are limited. Prototype does not exactly have the exact logic as compared to the actual software application. Prototype is used to allow the users to have a feel of what or how it is like and having to see the developer proposals before the actual implementation. Having a prototype enables developers to further understand the requirement the clients may have which are user specific and may not have been addressed during the initial specification. These are the following steps on how to design a software prototype.

1. **Basic Requirement Identification**

Just like any process models, this step is very essential as this is the step where the developers identify the needs and certain requirements which they think is important for the software.

1. **Developing initial prototype**

This step is where the initial prototype is developed. It contains the basic requirements and as well as provided the user interface. The features that are showcased in the initial prototype may not appear as exact to the actual software that is to be developed. However, it gives the exact look and feel to the customer.

1. **Review of the prototype**

The initial prototype is then presented to the customer and the other important stakeholders that is involved in the project. They, the developers then gather the feedback they get from them and after, they use the feedback that they have receive to further enhance the product which is under development.

1. **Revise and Enhance the Prototype**

The feedback is further discussed and taken into consideration at this stage. This stage is where amendments are done accordingly. Also, this is where negotiations occur between the developer and the customer. Negotiation factors are examples like time, budget constrain and technical feasibility of the actual one. the changes are then made and a new prototype is developed. This prototype is then showcased to the customers again and the cycle repeats until all the customer’s expectations are met.

There are different types of software prototypes used but I will be explaining on the following:

* **Throwaway/Rapid Prototyping**

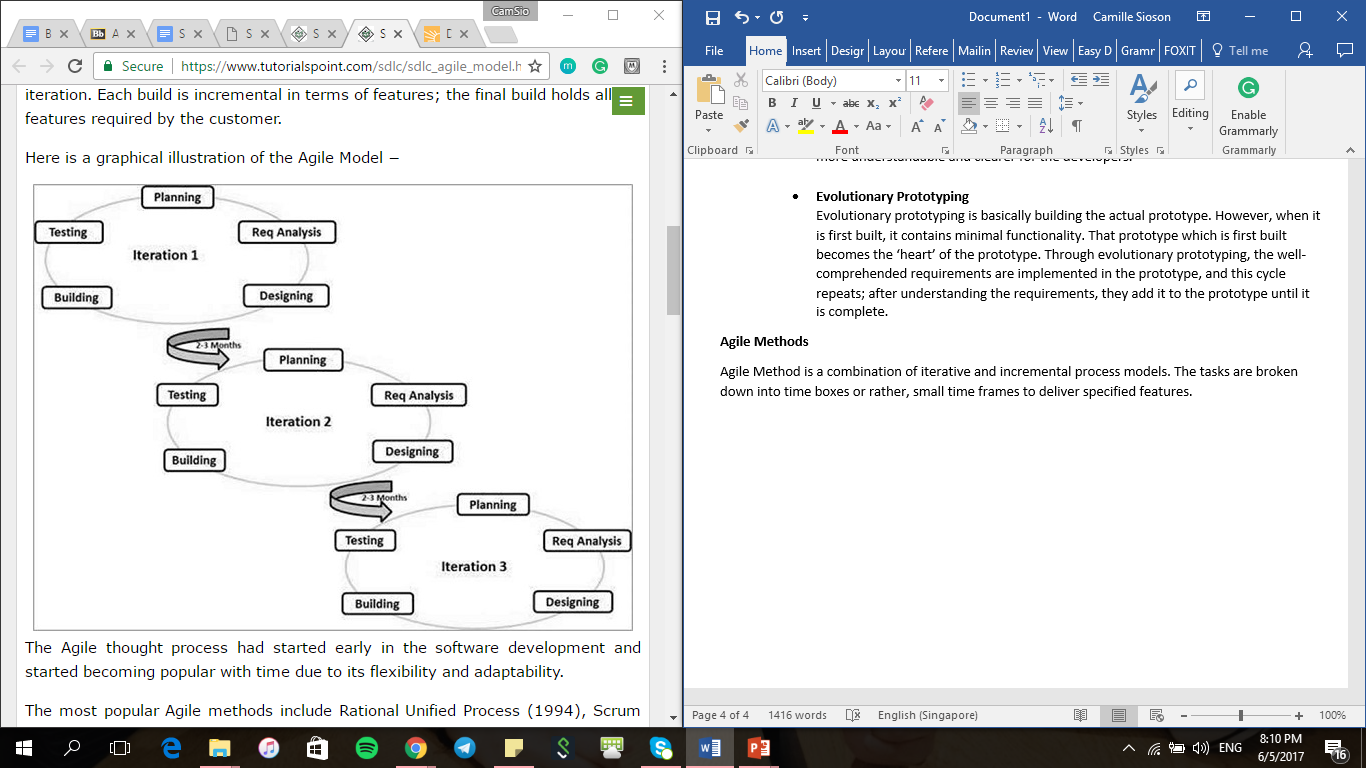
During initial specifications, there will be specifications for the software which contains requirements that are hard to comprehend. Therefore, a throwaway prototype implements those requirements that are hard to comprehend and is built.

After which, they will show their client and after validating with the client and clarifying the specification, they then throw away the prototype and a full-scale system is then built based on the specification they have gathered which is now more understandable and clearer for the developers.

* **Evolutionary Prototyping**

Evolutionary prototyping is basically building the actual prototype. However, when it is first built, it contains minimal functionality. That prototype which is first built becomes the ‘heart’ of the prototype. Through evolutionary prototyping, the well-comprehended requirements are implemented in the prototype, and this cycle repeats; after understanding the requirements, they add it to the prototype until it is complete.

**Agile Methods**

Agile Method is a combination of iterative and incremental process models. The tasks are broken down into time boxes or rather, small time frames to deliver specified features.

The picture above illustrates the Agile Method. As mentioned, the tasks are broken down into small time frames and this time frames must be the same for all iterations. At the end of the iteration, a working product should be showcased. There will be 10 iterations and each iteration, the customer will be shown a working software that is enhanced and updated with the features for that iteration and if there are any suggestions or changes the customer would want to make, they can then apply the changes for the next iteration.

## **Raeburn Timothy Nigel (1506381A)**

**What is Software Development Life Cycle(SDLC)?**

Software Development Life Cycle(SDLC) is used to describe what goes on from the start of a project to deployment of software system. SDLC are made up of different phases. Each phase plays an important part of SDLC as it ensures success of the software system and satisfaction of its users.

The 5 phases of SDLC are

* Requirements Engineering
* Analysis & Design
* Implementation (coding)
* Testing
* Deployment

Software Development Life cycle (SDLC) is different from Software Life cycle (SLC). For SLC has six phases which is the 5 phases from SDLC and the extra phase that is software maintenance. Software maintenance checks for bug fixes and does enhancement of the software.

**Software Development Models (Process Model)**

For Software Development Life Cycle (SDLC) there are process models that are used to plan steps to take to develop the project. The process model is used to predict what that will be done and to make adjustment or changes to the project and analyze the current development process.

The process models are

* Waterfall Model
* Prototyping
* Unified process

**What is a Waterfall Model?**

In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

**Advantages of using Waterfall Model**

* Easy to control and maintain as it completed one at a time
* Each task produces results and the process are well documented
* Complex task is divided into smaller task which make it more manageable
* Suitable for smaller projects where user requirements are understood clearly

**Disadvantages of using Waterfall Model**

* Cannot accommodate changes if user would like to change requirements
* There high amount of risk and uncertainty as product can only be seen at the end and no opportunity to clarify user requirements at the early stage
* Not suitable for projects where requirements have higher chance of changing by the user.

**When it useful?**

Waterfall Model is useful for smaller project as the requirements is understood clearly and when the projects have no changes

**What is Prototyping?**

Developer will start with a prototype of the system first instead of developing a complete system. Prototype is for the developer to clarifying requirements and it can also be used to demonstrate to the user, design the user interface and to verify that the new system is working.

For prototype that are two types which are throwaway or evolutionary prototyping.

**Throwaway Prototyping**

Throwaway prototyping is for the user to try out and evaluate. Once the user provided their feedback it will be added into the development of the system. The objective is to ensure the user requirements are understood clearly. The advantages are the satisfaction of the user as the developer will check with user and if changes are made prototype will be remade. The disadvantages are all the hours putting together the throwaway prototypes.

**Evolutionary Prototyping**

Evolutionary Prototyping is where the prototype is shown to the user where suggestion for improvements and feedback are given. The objective is for the developer to design and build a prototype and show the user for feedback, once feedback is given the developer will refine the prototype and when refine is finish the final product is made. The advantages are the user engages with the system and the user requirements are most likely met. The disadvantages are knowing when to finish the development and stop changing the system.

**Advantages of Prototyping**

* User requirements are well understood
* Problems can be identified during the early stage
* User feel involved with development of the system

**Disadvantages of Prototyping**

* Changes may cause the software structure to be corrupted.
* Building prototype cost money possible hardware and development time. The real system is being on hold as the prototype are being worked on
* There may be user confusion as some features appeared during the prototype which are removed during the final product which may cause user to feel disappointed

**Why is it useful?**

Prototyping is useful as requirement are understood clearly and if there any problem it can be identified during the early stage to prevent further project risk.

**What is Unified Process (UP)?**

Unified Process(UP) is a traditional view of system implementation as it seen as a series of steps toward implementation. Unified Process is made up of 4 phases which are Inception, Elaboration, Construction and Transition.

**Meaning of each phase**

Inception – Feasibility study to establish whether the proposed system is worthwhile.

Elaboration – List of requirements from the user, design architecture and development of the project plan that include project risk

Construction – Development of the system

Transition – Putting the system to use



UP uses an iterative fashion to accommodate ongoing risks. Each phase consists of one or more iterations and each iteration consist of disciplines such as analysis, design, testing and etc.

**Advantages of Unified Process(UP)?**

* Able to resolve project risk that with user additional requirements for changes or additional request
* As the up process goes on there less need of integration as the integration process has been going on throughout the development process

**Disadvantages of Unified Process(UP)?**

* When doing UP the developer need to be an expert
* The integration that going on throughout process of development may cause confusion that may add more problems during the testing stages

**Why is it useful?**

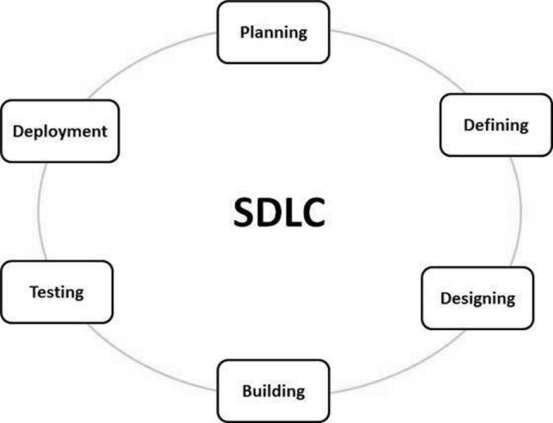
Unified Process is useful for bigger project as if the user decides to change the requirements UP is able to resolve the project risk.

## **Muhammad Nur Afnan Bin Abdul Rahim (1502527C)**

**Definition of Software Development Life Cycle(SDLC)**

It is a conceptual model that describe the phases involved in the software development process. Each phase is required to produce results that are necessary for the following phase. It is a structure that is mostly used by the development team in the software organization. It includes the plan of the development, maintenance, replacement and enhancement of a specific software. Methods that are developed by SDLC aims to improve the quality of the software and the development process.

**Phases**

****

The Software Development Life Cycle(SDLC) consist of different phases such as:

1.Requirements engineering (Planning)

* Plans the approach of the project and conduct findings on the user’s requirement and the level of risk that is involved during implementation of the project.

2.Analysis(Defining) & Design

* The documentation of the requirements of the project for the approval of customer or market analysts.
* A mock up design of the entity which will be developed for the project.

3.Implementation (coding)

* Development through coding that is based on the detailed design.

4.Testing

* Ensuring that the software or product is reliable with testing.

5.Deployment

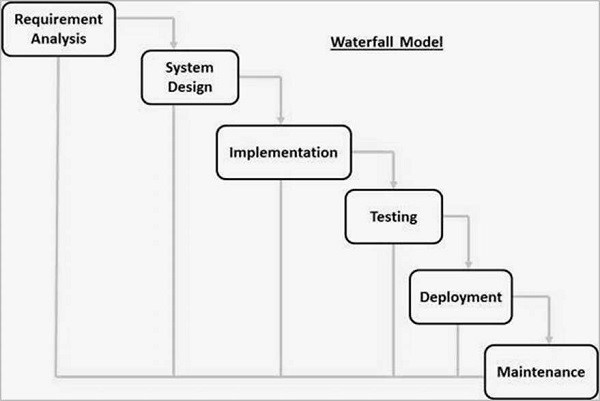
* Delivering the application to the customers.

**Software Development Models**

There are different software development models that are defined and designed which is used for the software development process. This includes the following:

**Waterfall Model**

In the waterfall model, every phase is completed before proceeding to the following phase. The development process model works in linear sequential flow which means that any phase can only being if the previous phase has complete. The outcome or final product of the phase acts as the input for the next phase. With the waterfall approach, the software development process is separated into different phases.



**·       Requirement and Analysis**

Every requirement of the system is gathered and documented in the requirement specification document. Usually take about 1.5 months.

**·       System Design**

The requirements from the first phase are being analyzed before designing the system. The design consists of defining the hardware and system requirements. This phase usually takes 2 months.

**·       Implementation**

The system is implemented based on the system design that is drawn from the previous phase. This phase usually takes 4 months.

**·       Testing**

The system is being tested for any faults or failures occurred. This phase usually takes 2 months.

**·       Deployment**

The system is being deployed to the consumer market or customer environment.

**·       Maintenance**

Maintenance of the system such as patches released to fix any issues. Features of the system are being updated too.

**Application**

Waterfall model is mostly used in scenario such as:

* Smaller scale project that can be completed in a short period.
* Product requirement and definition is clear and stable.

**Benefits**

* Deliverables are being clearly defined at each phase.
* It is easy to control as only one activity is handled at a time.

**Drawbacks**

* There is a higher risk involved during the project.
* Unable to accommodate changes to the requirements.

**Unified Process**

The Unified Process(UP) uses an iterative approach to deal with ongoing risk. It is realized by the creators of UP that specifying all requirements completely and accurately before the analysis and design phase is an impossible task. Thus, each phase in Unified Process can interact with the previous phase. At each iteration, modification and addition of new functionality is added. Unified Process requires prioritizing the highest risk at each phase of the process. The different phases that are in Unified Process are:

**·       Inception**

Feasibility is studied, project requirement is collected, project scope is understood and use cases are drawn out.

**·       Elaboration**

The large portion of the system requirements are being determined with process such as use case diagram, conceptual diagram, design architecture and project plan that are being developed.

**·       Construction**

Software engineers developed the system by making use of the developed requirements, use cases and system architecture.

**·       Transition**

The system is being deployed to the customer with feedbacks included.

**Application**

This software development model is mostly used in situation in the following scenario:

* The requirements of complete system are defined clearly
* New technology is used by the development team when working on the project.

**Benefits**

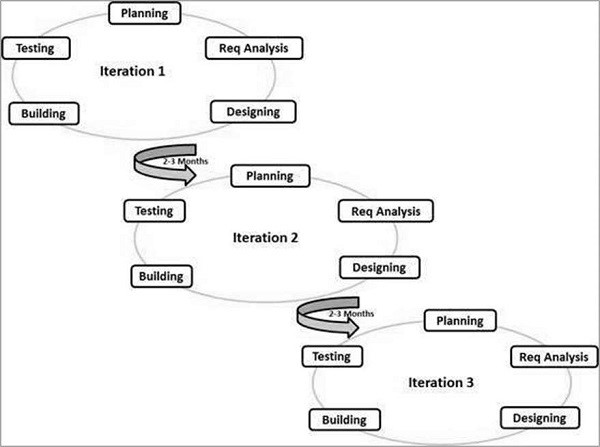
* Unified Process allows the reuse of components which reduce the development time of the project.
* It is proactive in resolving the potential risk involved with the project.
* It makes the process of testing and debugging easier due to small iteration.
* Suitable for large scale and mission-critical projects.

**Drawbacks**

* The members of the project should be knowledgeable in their field to develop a software under this technology.
* Development process is complex and disorganized.
* System architecture may cause problems as not all requirements are gathered at the beginning stage of the life cycle.

**Agile model**

The Agile model handles different projects differently and the existing method should be modified based on the project requirements. The tasks in Agile model are divided into smaller time frame which focus on specific features. Like the Unified Process, Agile model also uses the iterative approach and software build are delivered after each iteration. The same duration of time is applied for all iteration. Agile method focuses on the basic core features that are required for the product and continue to develop other features for the following iterations. If there are remaining features that are not delivered, it will continue to be developed for the subsequent iteration based on the priority of the features. Each software builds increment features which will meet the requirement of the customer at the end of the build.



The picture illustrates the process of the Agile model.

**Principles of Agile model**

**·       Individual and Interaction**

During the development process, self-organization and interactions are important.

**·       Working software**

Instead of documentation, using a demo working software allows the customer to understand their requirements with communication.

**·       Customer collaboration**

Unlike other models, not all requirements are being gathered at the beginning and thus active interaction is crucial to get the right requirements.

**·       Responding to changes**

It focused on the adaptability to the changes during development.

**Agile model vs Non-Agile model**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Agile** | **Non-Agile** |
| **Approach of the methodology** | It uses adaptive software development method. | It uses predictive method that is not flexible to changes in the project. |
| **Iterations/ cycles** | Involves many iterations which is known as Sprints. | Cycles are limited |
| **Scale of project** | Mostly used in smaller scale project. | Mostly used in larger scale project. |
| **Documentation** | It requires lesser documentation. | It requires more documentation. |

**Benefits**

* Promotes strong team work.
* Customer’s feedbacks can be accommodated to the features and they are constantly updated of the working features based on the requirements.
* It only requires little or no planning.
* Changes in the requirements can be accepted even in the later phase of the development process.

**Drawbacks**

* Unsuitable for handling complex dependencies.
* Transfer of work responsibilities may be difficult due to lack of documentation.
* Increase the risk of sustainability, maintainability and extensibility.

## **Dixon Low Yi Sheng (1502015I)**

The software development life cycle (SDLC) is a framework defining tasks performed at each step in the software development process. SDLC is a structure followed by a development team within the software organization. It consists of a detailed plan describing how to develop, maintain and replace specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. Three software development models are Waterfall Model, Prototyping Model and Unified Process Model.