**Temasek Polytechnic**

**School of Informatics and IT**

**Diploma in Information Technology (IT)**

Software Development Life Cycle

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Mr Mel Goh |
| **Class** | P02 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| 1500564H | Shawn Yang |
| 1502728G | Xavier See |
| 1500340C | Yeo Kai Sheng |
| 1500858D | Yeo Ai Ling |

Table of Contents

[1. Shawn 2](#_30j0zll)

[1.1. Description: 2](#_1fob9te)

[1.2. Waterfall Model 3](#_3znysh7)

[1.2.1. Application 3](#_2et92p0)

[1.2.2. Advantages & Disadvantages 3](#_tyjcwt)

[1.3. Iterative Model 4](#_3dy6vkm)

[1.3.1. Application 4](#_1t3h5sf)

[1.3.2. Advantages & Disadvantages 5](#_4d34og8)

[1.4. Spiral Model 6](#_2s8eyo1)

[1.4.1. Application 6](#_17dp8vu)

[1.4.2. Advantages & Disadvantages 6](#_3rdcrjn)

[2. Xavier 8](#_lnxbz9)

[2.1 Software Developmental Life Cycle (SDLC) 8](#_35nkun2)

[2.2 The phases of SDLC are: 8](#_1ksv4uv)

[2.3 Waterfall Model 8](#_44sinio)

[2.4 V-Model 9](#_2jxsxqh)

[2.5 Agile Model 10](#_z337ya)

[3. Kai Sheng 11](#_3j2qqm3)

[3.1 Software Development Life Cycle (SDLC) 11](#_1y810tw)

[3.2 The phases of SDLC are: 11](#_4i7ojhp)

[3.3 Software Development Models 12](#_2xcytpi)

[3.3.1 Waterfall Model 12](#_1ci93xb)

[3.3.2 V-Model 12](#_3whwml4)

[3.3.3 Agile Model 13](#_2bn6wsx)

[4. Ai Ling 14](#_qsh70q)

[4.1 SDLC Definition 14](#_3as4poj)

[4.2 Software Development Models 15](#_1pxezwc)

[4.2.1 Waterfall Model Design 15](#_49x2ik5)

[4.2.2 V Model 16](#_2p2csry)

[4.2.3 Big Bang Model 17](#_147n2zr)

[5. References 18](#_3o7alnk)

# Shawn

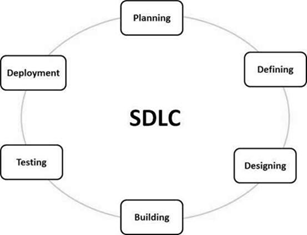
A short description to explain the meaning of Software Development Life Cycle (SDLC). Do ensure that you include three (3) software development models in your explanation. [5 marks - Individual]

## Description:

SDLC is a process followed during a software project. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of a specific software and the overall development process. The SDLC consist of the various phases:

* Planning
* Defining
* Designing
* Building
* Testing
* Deployment

The life cycle is basically the backbone of a software as it involves so many aspects of a software. As part of the SDLC methodology, current software systems are to be evaluated and software problems are identified. This could be performed by understanding user’s opinions when using the system and consulting with IT professionals.

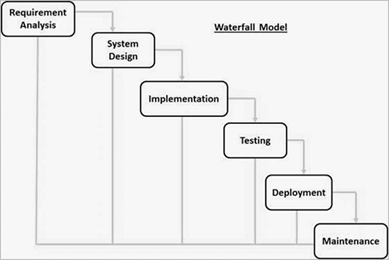


(Image Source:<https://www.tutorialspoint.com/sdlc/sdlc_overview.htm> )

SDLC Models

* Waterfall Model
* Iterative Model
* Spiral Model

## Waterfall Model



(Image Source:<https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm> )

The waterfall model as seen above is in sequence and linear form. It is very easy to perform and execute this lifecycle as each phase would have to be executed sequence before the next step could be completed. The waterfall model was the first ever SDLC method to be used for software development.

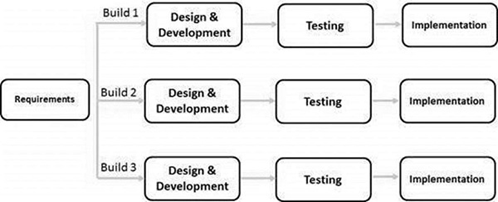
### Application

* Software development requirements are carefully analysed, detailed and fix.
* The definition of the product is well defined and in stable condition
* The technology used in this model is well known, uncomplicated and uncomplex.
* No ambiguous requirements are part of this model as this model requirements are clear and specific.
* The project does not take a particularly long duration to be completed.

### Advantages & Disadvantages

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * Software that is being developed using the Waterfall Model is well controlled. * The various stages of the Water Model can be provided with various deadlines as part of a larger schedule to complete the product development * This model is very useful for small managed projects where specifications are well understood and well defined. * Tasks provided for the various stages are easy to be arranged | * Software that is being developed using the Waterfall Model would not have much additional time for revisions and changes. * Once a software application reaches the testing stage, it is very troublesome for software developers to go back to previous stages to make changes * The Waterfall Model is not useful for larger, complicated and object-oriented projects. * The Waterfall Model is unable to accommodate software development specification changes during the various stages of the entire process. |

## Iterative Model



(Image Source:<https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm> )

The Iterative Model process begins with a small set of software development specifications and continues to be iteratively improved upon until they evolve into a complete system and becomes ready to be rolled out to be purchased in the consumer market.

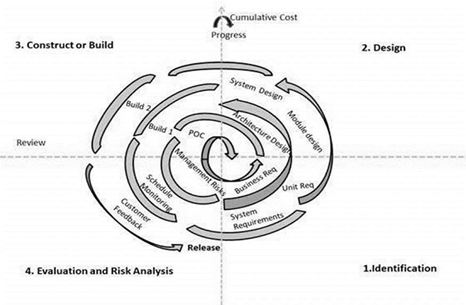
### Application

* Requirement specifications of the complete software system are defined clearly and understandable
* Key specified requirements are to be clearly understood and well defined when using the Iterative Model
* A duration is to be carefully set for market constraints
* Resources with needed specialised skill sets would not be available and would only be allowed in a contract context for specified iterations
* With the use of this model method high risk level features and goals are also involved that may change during later days

### Advantages & Disadvantages

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| * The model method produces a working model of the software that is being developed at an early stage of the model phases which makes it easy to detect problems * Because of finding problems at such early stages, software developers would be able to correct these problems quickly * The model method allows results of the development to be obtained at an early stage and periodically * · During the various phases of the model, progress during these phases can be monitored and examined | * This model method can only be used for complex and complicated software development projects * The reason is because small software system cannot be further broken down into smaller modular software’s * For this model method to be executed an additional amount of resources may be needed * For this model method to be successfully performed and executed, careful management observation is necessary |

## Spiral Model



(Image Source:<https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm> )

This SDLC model has a spiral shape to it as shown above and combines the design of iterative software development with the controlled and sequential aspects of the Waterfall Model. The combination between the two models would result in a Waterfall Method that would come with highly prioritising risk analysis.

### Application

* This SDLC model is widely used in software development market as it can easily synchronise with the development process of any product in the market
* The Spiral Model method is especially used when budget constraints and risk evaluation are regarded with priority
* The Spiral Model method is also used in a situation where the project development demands long-term commitment due to likely changes to economic targets as requirement specifications change overtime
* The Spiral Model method is also used in a case where customers are unsure about their software requirement specifications
* The Spiral Model method is also used in a case where the software requirement specifications are complex and need evaluation to get clarification

### Advantages & Disadvantages

|  |  |
| --- | --- |
| * Advantages | * Disadvantages |
| * Changing software requirement specifications during the process can be accommodated for * The Spiral Model method also accommodates for the heavy use of prototypes * The Spiral Model method also allows software requirement specifications to be stated more accurately and carefully. * The Spiral Model method also allows customer to take a peak of the software in its early stages of development | * The use of this SDLC Model would result in managing a project using this model being more difficult * The use this SDLC Model for a project would make it difficult to predict at early stages when the project would end * This SDLC Model overall has a complicated and complex process * This SDLC Model has an overall process that may go on and on indefinitely |

# Xavier

## 2.1 Software Developmental Life Cycle (SDLC)

Software Developmental Life Cycle (SDLC) is a process whereby software industries use to design, develop and test high quality softwares. SDLC describes a process of planning, creating, testing and deploying an information system. The life cycle defines a methodology for improving the quality of software and the overall development process.

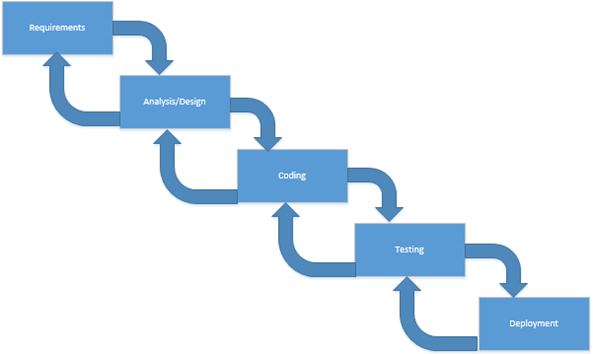
## 2.2 The phases of SDLC are:

1. **Requirements engineering** – Requirement gathering phase where one is need to find out what the client or customer wants the software to do. Expectations needs to be spelled out in detail and documented for the team or sponsors to see.
2. **Analysis and Design** – Analysis phase where a combination of text and diagrammatic forms to depict the requirements. Afterwards, produce a representation of an entity that will be built later on. The representation should include architectural design, user interface design, database design, etc.
3. **Implementation** – The coding phase where planned detailed designs will be converted into instructions written in programming language
4. **Testing** – Different type of testing will be performed including integration, system and user testing. Testing is to also ensure the system meets to the users’ expectations as mentioned in the requirements engineering phase. Errors and defects may be found and more work has to be done to correct it.
5. **Deployment** – Once all the testing has been completed the software will be deployed and distributed among a group of selected customers prior to the official release.

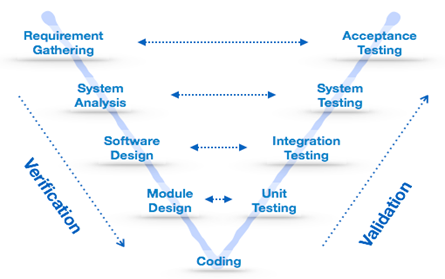
3 Examples of SDLC models are Waterfall Model, V-Model, and Agile Model

## 2.3 Waterfall Model

Waterfall model is where the phases of SDLC will function in a linear manner, moving on to the next phase after the first phase is finished before moving on. The model assumes that everything will go accordingly and there is no need to look back. This may cause serious problems if the previous issue was unsolved/undiscovered or if a new issue arises that needs to be solved in the previous phases.



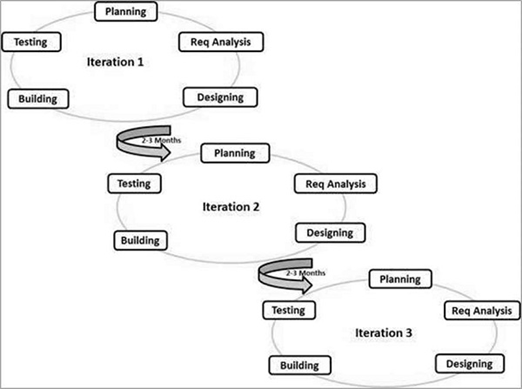
## 2.4 V-Model



The V-Model unlike the Waterfall Model, provides the means of testing of software at each stage in a reverse manner. It is able to go back to the previous phase compared to Waterfall Model where once it is finished there is no chance to change.

The V-Model is also known as verification and validation model as both verification and validation go in parallel. Test plans and test cases are created to verify and validate the product according to the requirements of that stage. For example, in requirement gathering stage, the test team will prepare all the test cases in correspondence to the requirements (as see in the diagram above). When the product is developed, and is ready for testing, the test cases prepared will verify the software against its validity towards requirements at the current phase.

## 2.5 Agile Model



The Agile Model is a process where every project needs to be handled differently and the existing methods needs to be enhanced to best suit the project requirements. Agile Model is based on the adaptive software development methods while the traditional SDLC models such as Waterfall Model is based on a more predictive approach. There are no detailed planning but has clarity on what features are needed to be developed. It is a feature driven development and the project is adaptable throughout. It is a combination of iterative and incremental process models with the focus on process adaptability and customer satisfaction.

# Kai Sheng

## 3.1 Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a series of steps, or phases that provides a model in the software development process. The life cycle defines a methodology for improving the quality of software and the overall development process

## 3.2 The phases of SDLC are:

Requirements gathering – This phase is essential for the project managers and stakeholders. The purpose of this step is to find out the scope of the problem and determine solutions. Resources, costs, time, benefits and other items should be considered at this stage.

Analysis and Design – The second phase involves working on the source of their problem or the need for a change. In the event of a problem, possible solutions are submitted and analysed to identify the best fit for the goal of the project.

Implementation – In this phase, Coding will begin according to the requirements, analysis and design discussed in the previous phases. It is the primary focus for the developer and the longest phase of the Software Development Life Cycle.

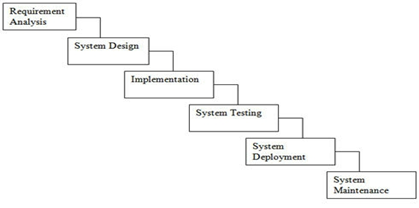
Testing – After the code has been developed, it is tested to make sure it meets the needs addressed and gathered, as well as finding defects or bugs. During this phase all types of [testing](http://istqbexamcertification.com/what-is-functionality-testing-in-software/) such as [unit testing](http://istqbexamcertification.com/what-is-unit-testing/), [integration testing](http://istqbexamcertification.com/what-is-integration-testing/), [system testing](http://istqbexamcertification.com/what-is-system-testing/), [acceptance testing](http://istqbexamcertification.com/what-is-acceptance-testing/) are done.

Deployment – Once the software has been successfully tested and no issues occur in the software, it will be deployed to the customer for their use.

Maintenance - Software maintenance is done for future reference, to boost performance, add new capabilities or meet additional user requirements.

## 3.3 Software Development Models

### 3.3.1 Waterfall Model

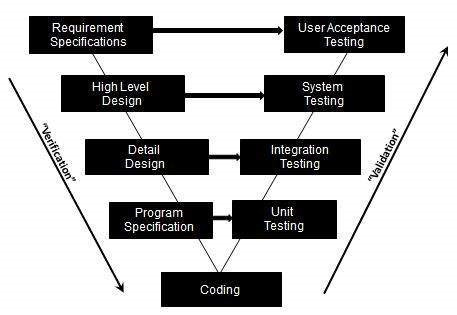


The Waterfall model, also known as Linear Sequential Life Cycle Model is the first SDLC model used in system engineering. the whole process of software development is divided into separate phases and each phase consists of series of tasks and has different objectives.

The Waterfall Model is very successful approach for the small projects where requirements are very well understood. and is simple and easy to understand and use. Additionally, as phases are processed and completed one at a time, phases do not overlap.

However, once an application is in the testing phase, it is very difficult to go back and change from the previous phases. As such, it is not suitable for project with high requirements and risk of changing.

### 3.3.2 V-Model

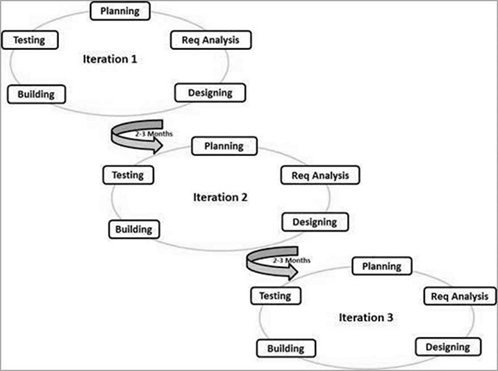


The V-model provides the association of a testing phase for each corresponding development stage. This means that for each phase used in the development cycle, there is a directly associated testing phase.

The V-model method is also very easy to understand and simple to apply. Furthermore, many testing activities are performed in the beginning which will result in less chances of defect or bug to be occurred at final testing phase.

However, it is not a good model for complex and object-oriented projects and it should be followed for the project where very less probability to make the changes in the middle of testing or phase which are unplanned.

### 3.3.3 Agile Model



The Agile model is a combination of model and document systems based of best practices. It breaks the product into small incremental releases with each release building on previous functionality. Each release is tested carefully to ensure software quality is preserved.

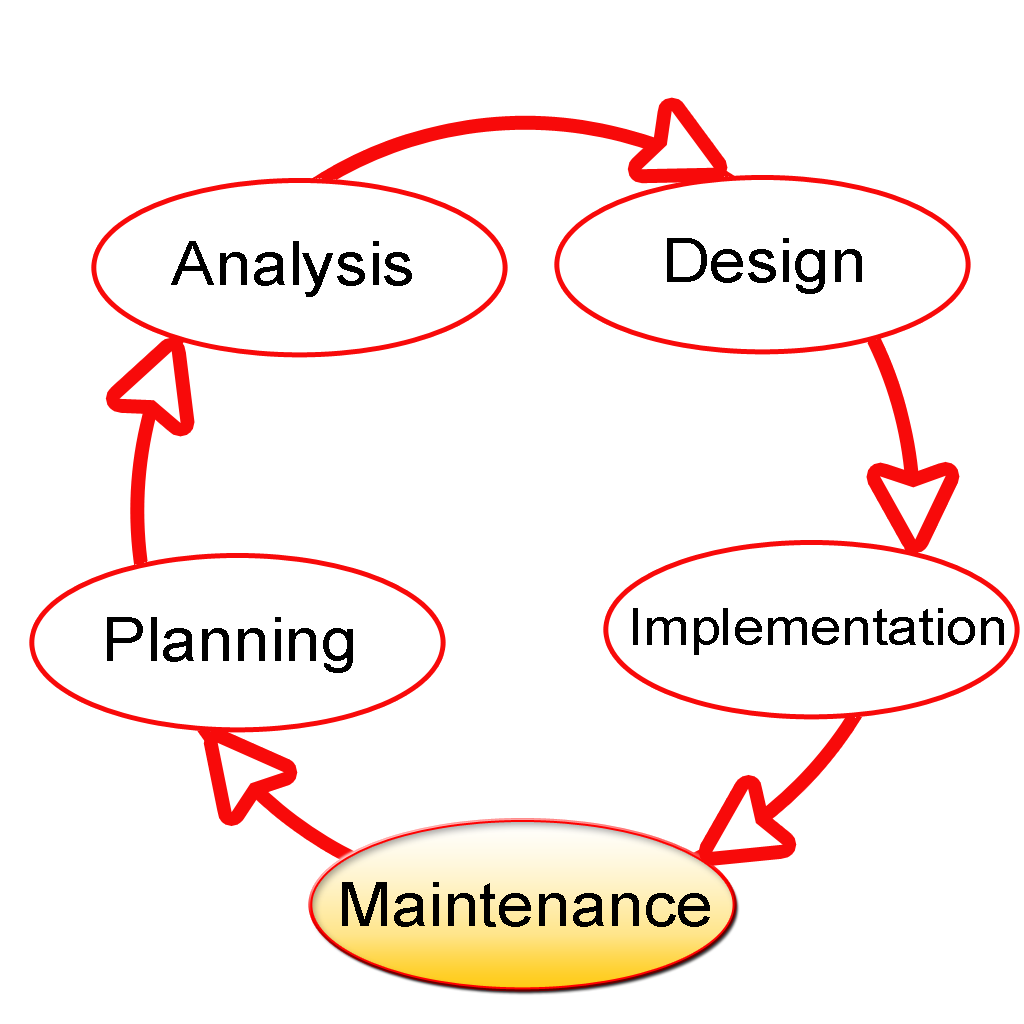
The benefits of using the agile model is it delivers early partial working solutions and it provides flexibility to developers as well as promote teamwork. Moreover, changes can be made whenever.

However, due to the lack of documentation, it will result in high individual dependency and transfer of technology to new team members or other department may be quite challenging.

# Ai Ling

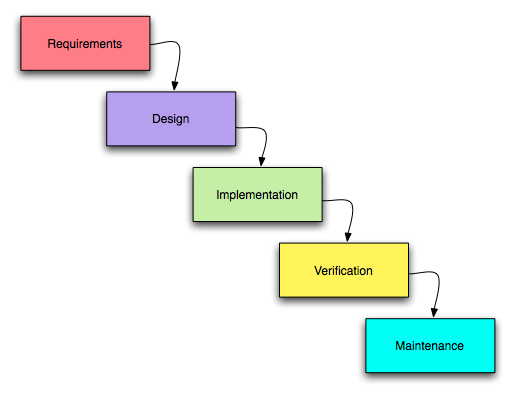
## 4.1 SDLC Definition

Software development life cycle(SDLC) is a term commonly used in software engineering, information systems and systems engineering to represent the action for planning, creating, testing and deploying an information system. It describes using a detailed plan, how to develop, maintain, replace and alter/enhance specific software(s). This life cycle defines a set of rules to improve the quality of the software and the overall development process.

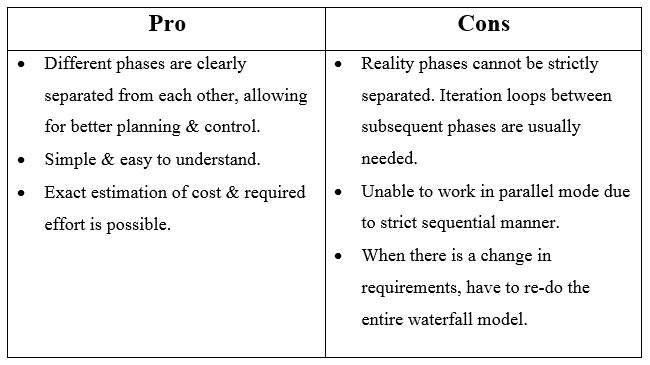


## 4.2 Software Development Models

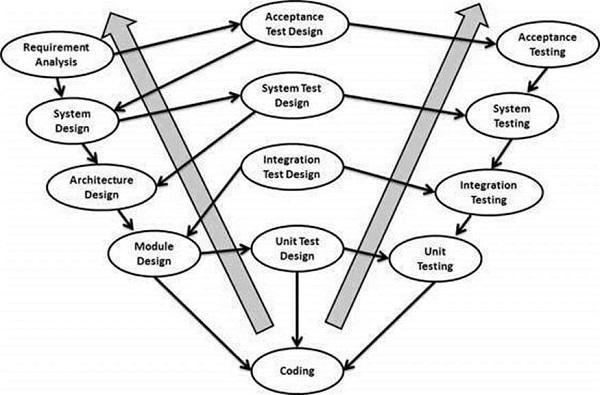
### 4.2.1 Waterfall Model Design



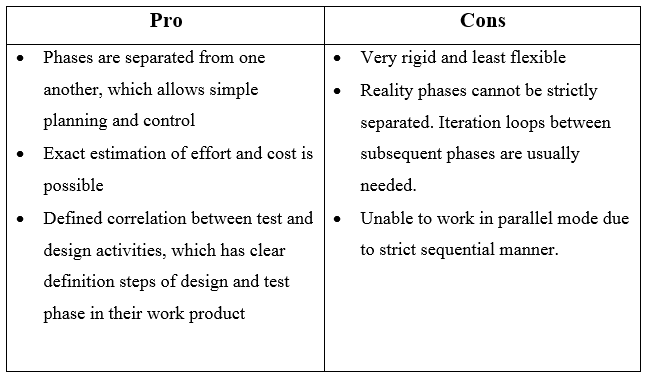
It is a linear-sequential life cycle model. It is very easy to figure out and apply. In a waterfall model, in order to move onto the next phase, the previous phase have to be completed as no overlapping in the phases. Each phrase of the waterfall model has a certain set of documents, documents at the front normally serves as inputs for later phrases. Waterfall is also the first SDLC model to be used frequently in Software Engineering to make sure that the project has a favourable outcome.



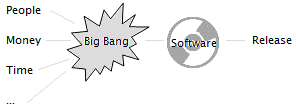
### 4.2.2 V Model



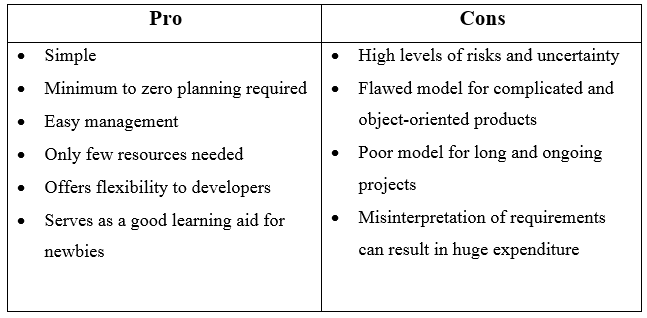
V-model is a SDLC model where execution of processes happens in a sequential manner in a V-shape, which is also called Verification and Validation mode. It is an extension of the waterfall model, however unlike waterfall model, it goes upwards at the coding phase to form a V shape, the reason for the line to head upwards is that the testing phase would agree to the design phase, which make it able to verify each of the design step individually.



### 4.2.3 Big Bang Model



It is an SDLC model where software engineer does not follow any specific process. The progress begins with the required money and efforts as the input, and the output is the software developed which might or might not be as the client demands. It does not follow and contain a small amount of planning needed. The client does not have an idea of what they are looking for and the necessity are start on the fly without much analysis. Normally this model is used for smaller projects where the teams are small.



# References

**Shawn:**

SDLC - Overview. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_overview.htm>

SDLC - Waterfall Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>

SDLC - Iterative Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm>

SDLC - Spiral Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm>

**Xavier:**

**Kai Sheng:**

SDLC - Spiral Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm>

SDLC - Waterfall Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>

Rouse, M. (2007, February). Waterfall Model. Retrieved from

<http://searchsoftwarequality.techtarget.com/definition/waterfall-model>

SDLC - V-Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm>

What is V-model- advantages, disadvantages and when to use it? (n.d.). Retrieved from

<http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/>

N. (n.d.). What is V Model in software testing and what are advantages and disadvantages of V Model. Retrieved from

<http://testingfreak.com/v-model-software-testing-advantages-disadvantages-v-model/>

SDLC - Agile Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm>

**Ai Ling:**

SDLC - Waterfall Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>

SDLC - V-Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm>

SDLC - Big Bang Model. (n.d.). Retrieved from

<https://www.tutorialspoint.com/sdlc/sdlc_bigbang_model.htm>