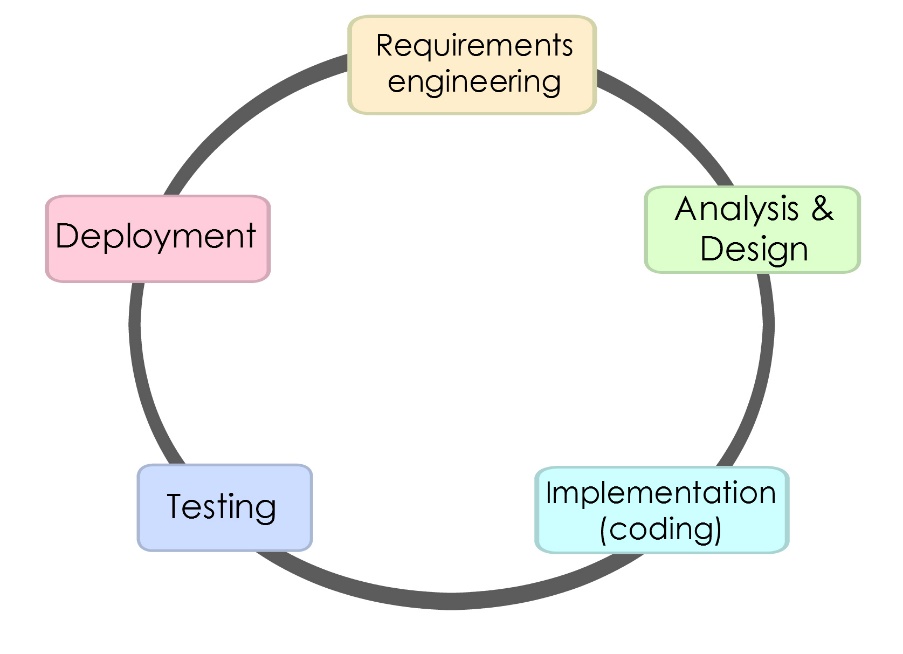
1. **A short description to explain the meaning of Software Development Life Cycle (SDLC). Do ensure that you include three (3) software development models in your explanation.   
   [5 marks - Individual]**Software Development Life Cycle (SDLC) is a series of phases used by software industries. The phase helps to design, develop and test a piece of software. It is aimed to provide customers with a high-quality software that they will be satisfied with.



SDLC consists of a detailed plan that explains how developing, maintaining, replacing, altering or enhancing a software works. For a typical SDLC, the life cycle defines a methodology to improve quality and overall development process.  
1. Requirements Engineering   
This is the first phase in the cycle which is the most important phase. At this phase, we will find out what the client expects the software to do. The inputs for this stage comes from customers, surveys, sales department and experts in this area. The information gathered will be used to plan the basic project approach. Feasibility study is being conducted as well to define various technical approaches that should be followed to implement the project with minimal risk.

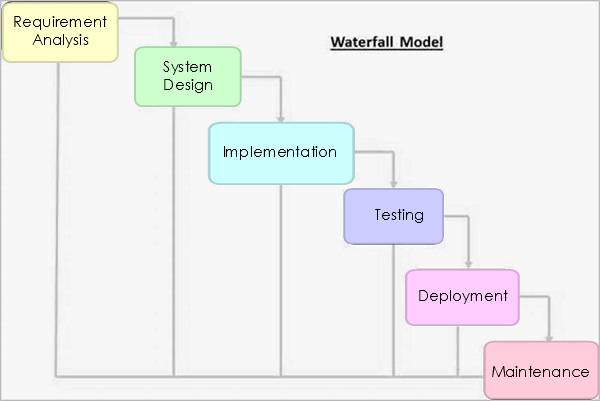
2. Analysis & DesignEnsures that initial security checklist and review is completed successfully during this phase. Analysis combines text and diagrammatic forms to portray the requirements while design produces an illustration that will be built later. Purpose of Design phase is to make the requirements into a detailed software.

3. Implementation (coding)After the design phase, the actual development starts and the product is built. This is why the design is completed in a very detailed manner, an organized design will allow the generating of codes to be done without much trouble. They type of programming language used is determined by the software that is being developed.

4. TestingMakes sure software meets customer’s expectation and needs. Although testing occurs at every stage of SDLC, this phase refers to stage where defects and errors are reported, retrieved and retested.

5. DeploymentWhen the software is thoroughly tested and is ready, a selected number of customer to trythe software before it has been released to the market. Some of the software might require prior training before getting started while some do not. Based on the feedback given by the selected few, the product may be released into the market or enhanced.

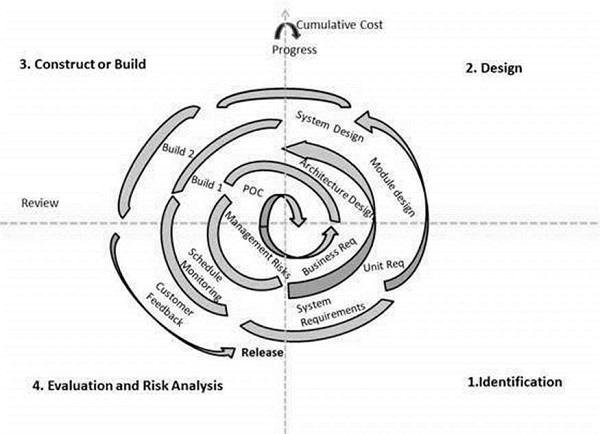
**Types of SDLC Models**SDLC Models are designed and followed during the development of software process. Each of these models have unique steps to follows to ensure the software development is a success.  
- Waterfall Model  
- Integrative Model  
- Spiral Model  
- V-Model  
- Big Bang Model  
  
1. Waterfall Model  
The first process model that was released. As the name suggest, the progress flow downwards just like how a waterfall work. Each phase must be completed before moving onto the next phase, not overlapping.



* Requirement Analysis: analyse all possible requirements that is to be developed and document it in a requirement specification document   
  (a description of a software system, includes functional and non-functional requirements)
* System Design: required specifications are analysed in this phase and the design will take place, defining the overall system architecture.
* Implementation: the development of software starts and is made in small programs named “units”, which will be tested.
* Integration and Testing: after testing, the units will be compiled and integrated into a system. It will be tested again for errors.
* Deployment of system: when the testing is done, the software will be ready to be released to the market.
* Maintenance: Patches are released to solve issues that came up in the client environment.  
    
  Pros:   
  - splits and divides complex task to make it manageable  
  - each task requires and will produce a clear deliverable

Cons:  
- software will only be seen at the end of project  
- problem that is discovered early cannot be solved first as there is no overlap

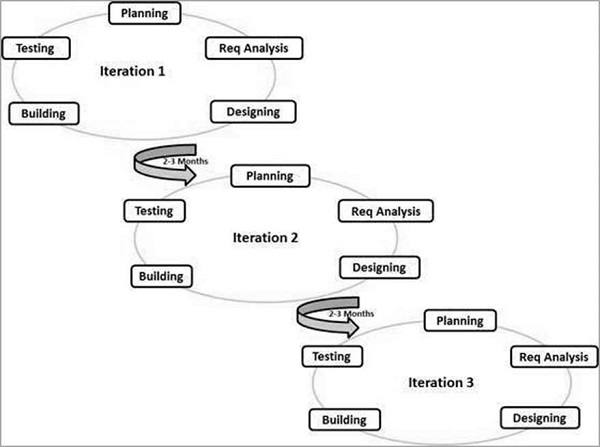
2. Spiral Model   
Something like the Waterfall model, the Spiral model is a repeated development of phases in Waterfall model. There are 4 phases in a Spiral model and when the software moves through these 4 phases, a spiral is formed.



* Determine objectives (Identification): this is where all the requirements are conceptualised. At the baseline of the spiral, business requirements are gathered. Identifying of system requirements, subsystem requirement and unit requirements are done. This phase requires the system requirements to be fully understood by continuously communication with the client and analyst.
* Design: this phase is where all the designing of software, architecture, logical, physical and final design are done in the subsequent spirals.
* Construct: this phase is where the software will be build and the Proof of Concept (POC) will be made to allow client to view and provide feedbacks.
* Risk Analysis: this phase will analyse the possible management risk and will be eliminated at this phase. (e.g. overbudget/time overrun)}  
    
    
    
  Pros:   
  - system can be seen clearly   
  - allows user of extensive prototypes  
  - development is split into smaller parts, making it easier to produce and eliminate risky parts that will be done earlier

Cons:  
- management and process of spiral model is more complex  
- not cost efficient   
- requires more documentation

3. Agile Model   
Unlike the previous 2 Models, Agile Model focuses on adaptability and customer satisfactory. This is done by combining iterative and increasing process model and rapid delivery of software. The software product will be split into small buildable products to be worked on. The final build consists of all the features required by the client.



Agile principles  
- Individuals and interactions: self-organization and self-motivations is highly required  
- Working software: to allow clients to fully understand clearly, the best way is to produce a working software. This will give them an idea of how it works instead of just reading words.   
- Interactions: requirements gathered in the start of the project may vary during the duration of the project. This is why it is important to constantly interact with the client  
- Responding to change: quick responses to change.  
  
Pros:  
- teamwork and cross training   
- requirements can be changed during the project phase  
- documents are easily made  
- easy to manage   
  
Cons:   
- A leader is required for project to carry out  
- not suitable for complex dependencies  
- rely on customer interaction. If client do not understand

Reference:  
<http://doit.maryland.gov/SDLC/Documents/SDLC%20Phase%2005%20Design%20Phase%20Multiple%20Custom.pdf>  
<https://www.tutorialspoint.com/sdlc/sdlc_overview.htm>  
<https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>  
<https://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm>