**Temasek Polytechnic**

**School of Informatics and IT**

**Diploma in Information Technology (IT)**

Software Development Life Cycle

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Qi Yu Tao |
| **Class** | P02 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1300647C** | **Aaron Lim Jie Kai** |
| **1306307H** | **Chew Zhi Yong** |
| **1302901B** | **Nurul Hidayah Hamzah** |
| **1301354E** | **Zhi Mai** |

**Nurul Hidayah Hamzah <1302901B>**

**Software Development Life Cycle (SDLC)**

A software development life cycle (SLDC) is a process consisting of steps, or phases, that provide a model for the development or alteration of software products. It is used by software industry to design, develop and test high quality software with an aim of meeting or exceeding customer expectations. The methodology within the SDLC process can vary across industries and organizations. It helps produce a product to be cost-efficient, effective, and of high quality. The SDLC will map a proper deployment and decommissioning of the software once it becomes a legacy.

**Phases in SDLC**

Stage 6

Stage 5

Stage 4

Stage 3

Stage 2

Stage 1

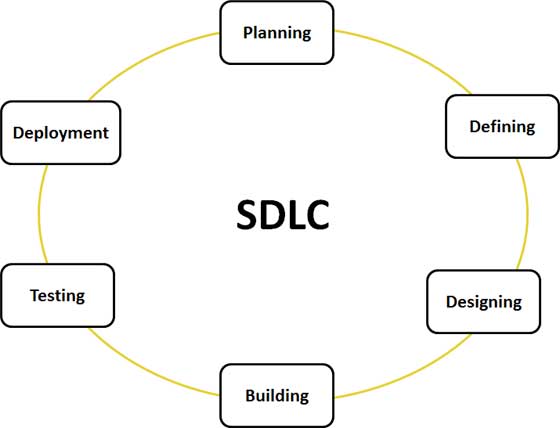


Figure A: Graphical representation of the various stages of a typical SDLC.

Stage 1: Planning and Requirement Analysis

Stage 2: Defining Requirements

Stage 3: Designing the product architecture

Stage 4: Building or Developing the Product

Stage 5: Testing the Product

Stage 6: Deployment in the Market and Maintenance

1. **Agile Model**

Agile model focuses on process adaptability and customer satisfaction by rapid delivery of working software product – combination of iterative and incremental process models. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications. It believes that every project needs to be handled differently and the existing methods needs to be tailored to best suit the project requirements. The tasks are divided to small time frames to deliver specific features for a release. Agile models are popular with managing time due to its flexibility and adaptability.

Developers will start off with a simplistic project design, and then begin to work on small modules. The work on these modules is done in weekly or monthly sprints, and at the end of each sprint, project priorities are evaluated and tests are run. These sprints allow for bugs to be discovered, and customer feedback to be incorporated into the design before the next sprint is run.

**Advantages:**

* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum.
* Suitable fixed or changing requirements.
* Working software is delivered frequently (weeks rather than months).
* Minimal rules, documentation easily employed
* Enables concurrent development and delivery within an overall planned context.
* Little planning required
* Easy to manage

**When to use Agile model:**

* When rapid production is more important than the quality of the product.
* When clients will be able to chance the scope of the project.
* When there isn’t a clear picture of what the final product should look like.
* When you have skilled developers who are adaptable and able to think independently.
* When the product is intended for an industry with rapidly changing standards.

1. **V-Model**

V-Model means Verification and Validation model. Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes. Each phase must be completed before the next phase begins.  Testing of the product is planned in parallel with a corresponding phase of development. This developmental process is balanced and relies on the verification from the previous steps before proceeding forward. So the cycle of the model has been divided into several phases and each one is supposed to yield a predefined product.

**Advantages of V-model:**

* Simple and easy to use.
* Specific goals are defined for each phase
* Focus on preparing test plans early in the process which give it higher chance to success
* Participate in the development and maintenance
* Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
* Proactive defect tracking – that is defects are found at early stage.
* Avoids the downward flow of the defects.
* Works well for small projects where requirements are easily understood.

**When to use the V-model:**

* The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
* The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.

1. **RAD Model**

RAD model is Rapid Application Development model. It is based on prototyping and iterative development with no specific planning involved. It uses minimal planning in favour of rapid prototyping. The functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. The products from RAD model can be developed faster and of higher quality through:

* Gathering requirements using workshops or focus groups
* Prototyping and early, reiterative user testing of designs
* The re-use of software components
* A rigidly paced schedule that defers design improvements to the next product version
* Less formality in reviews and other team communication

**Advantages of the RAD model:**

* Reduced development time.
* Increases reusability of components
* Quick initial reviews occur
* Encourages customer feedback
* Integration from very beginning solves a lot of integration issues.
* Easier to incorporate the changes within the development process

**When to use RAD model:**

* RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.
* It should be used if there’s high availability of designers for modelling and the budget is high enough to afford their cost along with the cost of automated code generating tools.
* RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).