SDLC Approach

Incremental Development

4/13/2022

Errors Team

**1. Incremental Development Approach**

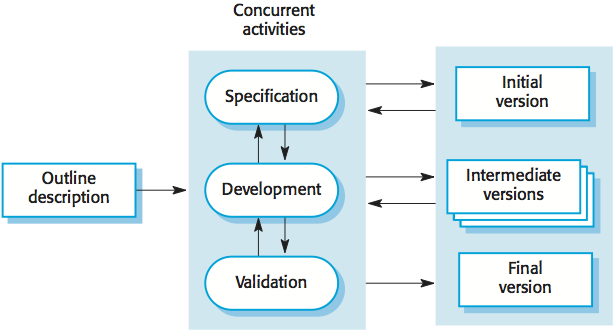
1.1. **Introduction**

When it comes to build a new software system, the project team starts to choose an appropriate SDLC approach depending on several factors such as time schedule, cost, clear user requirements, familiar technology, complexity, schedule visibility, advantages and disadvantages of each approach and how the approach can fit with the system nature. So accordingly, we have chosen incremental development approach.

1.2. **What is** **Incremental Development?**

Incremental development is a method of building software products in which a system is built piece-by-piece. The final requirement specification is clear from the beginning, and everyone knows the end result clearly. The system is broken down into small sub-systems which are designed, built and tested independently. This allows partial utilization of the product, but the full system isn't usable unless until the development is entirely done.

A good analogy to understand this model is looking at how a mason builds a wall. How the final wall should look like is already clear in their mind, and starting from zero they lay out the wall brick by brick. The wall becomes fully usable only when the construction is entirely done.



1.3. **Why Incremental Development?**

- We have chosen incremental approach as it is appropriate for a small-time range project and this in line with our project.

- Costs are reduced if we want to change something in the system according to customer requirements.

- We can keep up with customer easily to get his feedback on development work that has been done.

- It is possible to deliver and deploy useful software to the customer rapidly.

- It requires less analysis and documentation that has to be redone.

- Customers will be able to gain value and experience and use system early.

- Each increment is a prototype to help elicit for later increments.

-Lower risk of overall project failure.

1.4. **Incremental Approach Problems and Solutions**

- Problem: It is important to produce regular deliverables to measure progress.

- Solution: We intend to use version control system such as GitHub to keep information about each increment and to have the ability to produce deliverables regularly.

- Problem: System structure will be corrupted as new increments are added and software changes become increasingly difficult and costly.

- Solution: It is important to perform refactoring regularly to improve the software.

**2. Requirements Gathering**

**2.1 Functional Requirements**

**Functional Requirements**

**User requirement Definition**

1. The user wants to be able to create a new account and log in
2. The doctor wants to be able to publish an advertisement containing the specifications of the case he is looking for
3. The doctor wants to have the ability to search for a specific case that wants treatment, and selects from the results the case he is looking for and then sends him a request
4. The patient wants to be able to publish an advertisement containing the specifications of his condition to seek help from one of the doctors who are looking for cases to treat
5. The patient wants to be able to search the publications of doctors who are looking for patients to treat them, and send a request to the appropriate doctor
6. The user wants to review the requests sent to him and respond to them with approval or rejection

**System requirements specification**

1.1. The user wants to review the requests sent to him and respond to them with approval or rejection.

1.2. The user can create an account by entering his personal data.

1.3. The system must provide both the doctor and the patient with their own form through which they fill in their data

1.4. User logs in using email and password

2.1. The system must provide the doctor with the ability to publish a publication that includes the case data he is looking for, where he fills out the designated form with the details of the case and attaches relevant pictures if he wants

3.1. The system must provide a search box for the doctor so that he can write the keywords that help him to reach the case he is looking for

3.2. The system must contain a powerful search engine to reach publications that are similar to the keywords used by the user during the search

4.1. The system must provide the patient with the ability to publish a publication that includes a description of his condition, where he fills out the designated form with the details of the case and attaches relevant pictures if he wants.

5.1. Also, the patient will have the same ability as the doctor to search for advertisements, but the patient can search for doctors’ advertisements and vice versa

6.1 The system should provide the ability to review the requests sent to the user and the ability to approve or reject them.

6.2. The system must send a notification to the user when a request is received

6.3. The system must provide the ability to view the status of requests sent by the user to see if they have been rejected, approved, or are still pending.

**2.2 Non Functional Requirements**

**2.2.1. Performance**

* Ensure that only the necessary data is sent in the response to reduce the response time.
* Ensure that the server capabilities are compatible with the number of potential requests from users
* Compressing large data during the request and decompressing it during processing after the response.
* Keeping an empty amount of storage space to keep up with data growth.
* Back up data periodically

**2.2.2. Security**

* Ensure that password saved using hashing algorithm.
* Ensure that password rules are clear and strong.
* The system should use a security protocol when sending data over the internet.
* Ensure that each user can only access, manage the system according to his own restrictions (account type & ID) and permissions specified by the admin.
* The user’s account will be blocked after a number of failed attempts to login and a verification mail will be sent to user via the E-mail.
* There will be a verification email sent to users after registering for to ensure that there is no impersonation.

**2.2.3. Logs**

* Ensure that system have fully detailed action logging, with easily retrieval interface.

**2.2.4. Errors exception and handling**

* Everything that will happen in the system must be recorded in a deep report for the admin.
* Ensure that the system will keep record of every error.
* Ensure that error messages are clear to the user.
* Ensure that all the errors of the system will be handled by the admin.
* Ensure that standard error pages will appear in case of any errors.