PROJECT REPORT
ON

Software Development Lifecycle (SDLC) Analysis of Oracle Health

A comparative study of different models in relation to Oracle Healthcare System Development

Submitted To

NMAM Institute of Technology, Nitte

(Off-Campus Centre, Nitte Deemed to be University, Nitte – 574110, Karnataka, India)

On partial fulfilment of the requirements for the award of the

Degree of Bachelor of Technology

In

INFORMATION SCIENCE AND ENGINEERING

By

Cadlee Kriston D Souza

NNM23IS034

Under the guidance of

Dr. Jason Elroy Martis, Associate Professor, Department of Information Science and Technology, NMAM Institute of Technology, Nitte, Karnataka, India

PROBLEM STATEMENT

Your task is to analyse the software development lifecycle (SDLC) of a real-world system by conducting a comparative study of process models and their impact on requirements management.

1. **Case Study Selection**:
    - Choose an existing large-scale software system (e.g., an e-commerce platform, a healthcare management system, or an ERP system).
2. **Analysis and Comparison**:
    - Write a detailed report comparing **Incremental Development**, **Spiral Model**, and **Waterfall Model** for the chosen system. Highlight their suitability based on:
        - Functional and non-functional requirements.
        - Risk and change management.
        - Time and cost constraints.
3. **Requirements Engineering Process**:
    - Develop a simplified **requirements document** for the system, including functional and non-functional requirements.
    - Outline a strategy for **Requirements Validation** and identify potential challenges in this phase.

Table of Contents

Introduction

Oracle Health, a leading provider in healthcare information technology solutions and a known specialist in electronic health records (EHR) systems, was acquired by Oracle Corporation for USD 28.3 billion in June 2022, thus marking it as one of the biggest acquisitions in healthcare technology in recent history.

This project looks at analysing an EHR system developed by Oracle Health, and we're going to analyse its requirements engineering process and implementation strategies. The primary objective is to investigate 3 different development methodologies which are, The Waterfall Model, Spiral Mode and the Incremental Development Model.

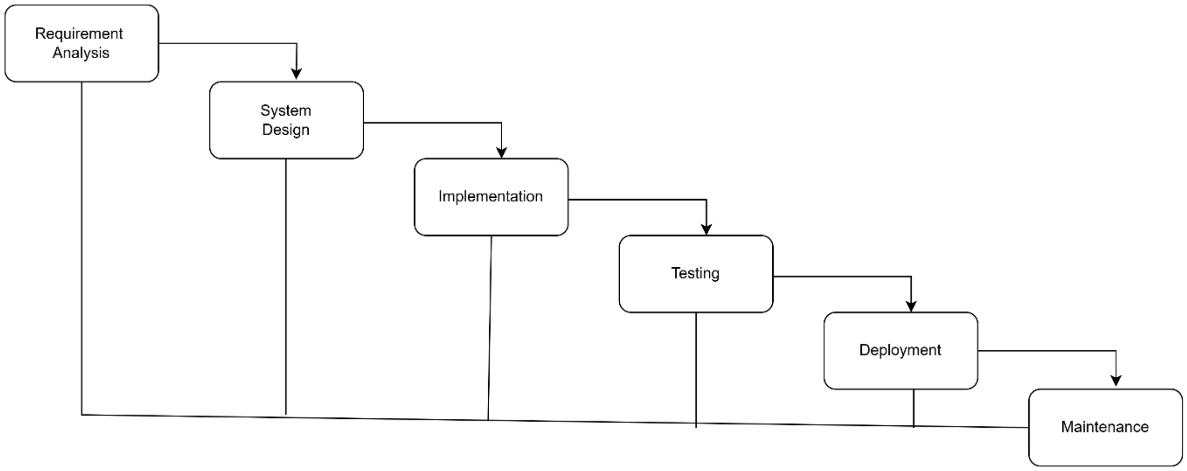
The several important aspects of software development will include the extended analysis of functional and non-functional requirements, risk management, change management, and time and cost constraints. This comprehensive analysis intends to figure out which development model would be most suited to fit healthcare's stringent parameters relating to patient safety and data security, system security, and reliability.

Health Management System requires particular attention for details, top security, and full integration ability. The choice of development methodology will greatly determine the success of the venture in question, especially in respect of attaining required governmental compliance and, mostly, patient safety. The research, therefore, will discern how the various development models address these challenges concerned with the implementation of any technological advancement in the health sector.

This report is most relevant given the trend of increasing digitization of healthcare services, developing a consolidated important role for effective, safe, and efficient Healthcare Information Systems in contemporary medical practice. The findings will provide further guidance into the best practices for software development and strategies for implementation within the healthcare sector.

Waterfall Model

The waterfall model which was first developed in 1970 is a linear and sequential approach to software development and it consists of various phases.



Let us see how our Oracle Health Model could implement the waterfall model.

i) Requirement Analysis

In this phase of the waterfall model we gather a lot of information such as requirements. We mainly have two types of requirements , Functional and Non-Functional Requirements. In this phase we need to make sure that the functional and non-functional requirements are well defined and in such a way that there is no confusion in between the two requirements.

ii) System Design

Once the Requirement Analysis and Gathering is done , we move on to the System Design Phase. Here we create a detailed system architecture design. This includes the database schema for patient records, user interface design and any other required modules or components. We also develop flowcharts, ER diagrams, and wireframes. The System designed should be in such a way that it should integrate with existing healthcare IT systems with ease.

iii) Implementation

In this phase the actual coding of the system takes placed based on the various design documents. Developers write the code for various modules like the patient record management, appointment scheduling etc. While coding we need to ensure that coding standards are followed. Here we also need to keep in mind of various features relevant to our

System which is adding,updating or viewing records. In this phase no changes can be done to the requirements which are already defined.

iv) Testing

In this phase the system is thoroughly tested to ensure it meets all the requirements which are already defined and is free or any errors. Here we develop various test cares based on our requirements definition, conduct performance test to see if the system can handle the expected load. And if there are any bugs or errors, it should be fixed before moving onto the next phase. We also need to conduct security testing to ensure compliance with HIPAA regulations.

v) Deployment

Now the system is ready to be deployed, we need to install the system on production servers, and also migrate data from exisisting systems if required. Should also train the users how to use our particular system, like healthcare providers,administrators, Doctors and Nurses.

vi) Maintenance

Here we need to keep updating the system to improve performance or to add new features. We also need to provide technical support to end users in case if they need any help or if they run into errors. In the medical field, as regulations can always keep on changing, we need to update our system to always comply with new regulations.

Suitability to Our System

i) Pros

- *Documentation & Compliance*
  The waterfall model emphasis on comprehensive documentation aligns perfectly with the standards in that of the medical field. Here each phase requires detailed documentation.

- *Quality & Safety*
  The sequential nature of the waterfall model supports the development of a reliable software which turns out to be crucial when dealing with patient data and medical decisions.

ii) Cons

- *Rigid Architecture*
  As healthcare technology is never constant and keeps rapidly changing, this model doesn't suit our model to constantly updating regulations or requirements once the development is done.

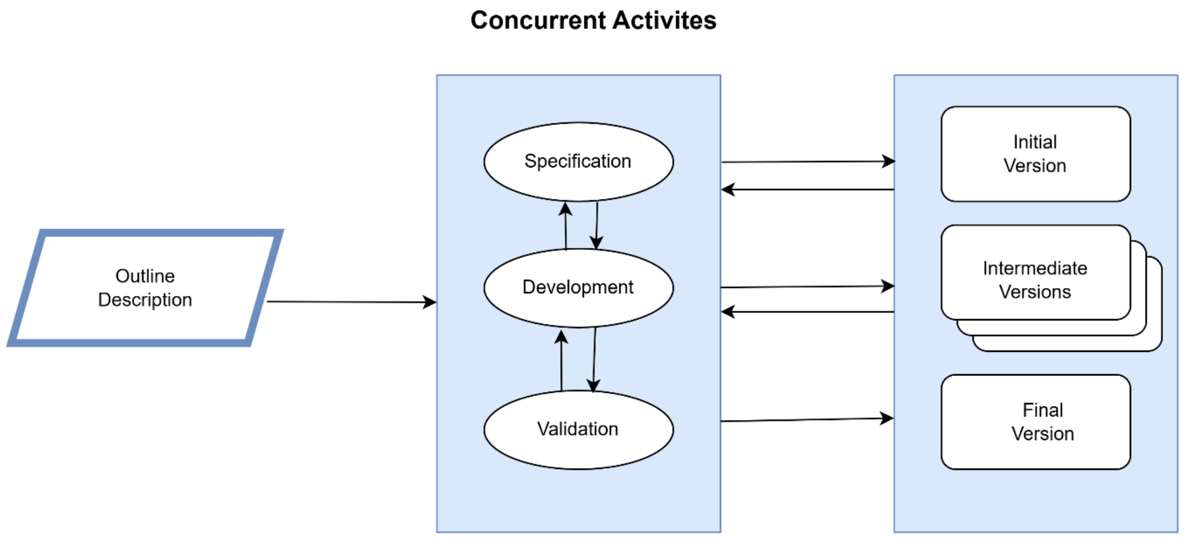
- *Cost*
  If requirements are changed once the deployment is done, implementing new changes can be very time consuming and expensive.

- *Integration Challenges*
  As modern healthcare systems require integration with various medical devices, this model makes it very difficult to adapt to integration challenges found later into the development phase.

Final Verdict

Our model doesn't suit the waterfall model because the healthcare requirements frequently change, and the waterfall model can not adapt to that. Healthcare professionals need to provide continuous feedback which is not possible with the waterfall model's late-testing approach.

Incremental Development Model.



Introduction

The incremental development model offers a systematic approach for developing of our system through various phases. In this model each phase builds upon the previous build/phase.

i) Outline Description Phase

Here we begin by establishing the foundational framework for the entire healthcare system. In this phase we define key requirements which includes patient management, electronic health records (EHR) , and various security compliance measures. This is one of the most important phases in this model as it sets a crucial step for the rest of the phases. Here the team creates a comprehensive roadmap prioritizing features based on the medical necessity and operational importance. For example, patient registration and basic medical records might take more importance rather than advanced analytical features.

ii) Specification Phase

Here detailed requirements for each phase are documented. For Oracle Healthcare, our first increment specifications might focus mainly on the basic patient management and the EHR functionality. Here each subsequent increment builds on the previous one with new features like billing systems. This phase includes detailed functional requirements, interface requirements with existing medical systems, technical architecture specifications, user interface requirements.

iii) Development Phase

Here the development phase includes important features in planned increments to set foundation to the Final version of this model. For example our model can be divided into 3 versions.

Initial Version – In this version we could focus on the basic patient registrations appointment scheduling, Fetching existing medical records from existing healthcare systems.

Intermediate Version – This version could focus on more advanced EHR functionality, integrating & advanced analysis of laboratory results and enhanced security measures.

Final Version – Here we could focus on how the Billing and Insurance processing works, advanced analytics, detailed reporting system with summaries and other AI Features and Administrative features.

iv) Validation Phase

The validation phase happens to every single version created by the model and not only on the final model, this ensures any fault or errors in the early parts of development can be fixed at an early stage and the error or fault should not be carried over to the subsequent versions. In this phase it undergoes functional testing by various healthcare professionals, security validation, integration testing with already existing systems, performance testing under various technical and medical scenarios and user acceptance testing in clinical environments to see how easy the system is for the end user to use on a day-to-day basis.

Suitability to our System

i) Pros

- Early Delivery
  Basic healthcare features are available quickly and we get early return on investment compared to other models.

- Incorporation of user feedback

As user validation is done at every step, any bugs or errors found at any increment can be fixed rapidly and continuous system improvement is observed.

- Flexibility
  This model can easily adapt to the fast-changing medical requirements and has the ability to adjust to new medical features.

ii) Cons

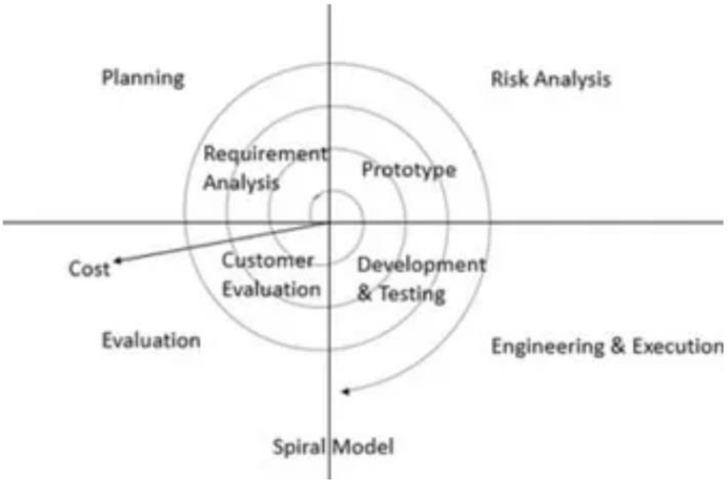
- Various Documentation
  In the model we will have various versions to maintain which requires extensive documents requirements.

- Resource Management
  We require constant team/support availability and a continuous stakeholder involvement.

Final Verdict

While the incremental model has a lot of benefits like rapid development and feature implementation, its limitations in risk analysis and security assessment makes it less suitable for a critical medical system.

Spiral Model



Introduction

The spiral model represents a risk-driven software development approach. It follows a distinctive spiral pattern where each loop represents a progressive phase of our system development with risk analysis at its core. This model is particularly suited for large models where risk management is crucial like our Oracle Healthcare System. While the spiral model in the diagram focuses more on requirements engineering, our description will cover the complete development lifecycle.

i) Requirement Gathering

The first phase of this model focuses on collecting & documenting business requirements through stakeholder consultations. Here we analyse the functional requirements, and the cost required. This phase is important as it makes sure that all stakeholders have a common understanding of the project

ii) Risk Analysis & Planning

This is one of the most important phases in this model. In this critical phase, potential technical and management risks are identified and assessed. Here the team plans strategies and allocates resources and has to evaluate alternate strategies.

iii) Engineering and Development

The development phase handles the system design, architecture planning, and actual implementation. Our team here works on prototypes to develop, implementation of codes and designs the database which will be used. This phase is used to deliver results through systematic development.

iv) Testing and Validation

Rigours testing ensures Oracle Heath's System quality through comprehensive testing with already existing healthcare system, user acceptance testing and performance evaluation under various scenarios. Security testing here focuses on HIPAA compliance and patient data protection & patient safety protocols.

v) Evaluation & Deployment

This phase involves thorough review by stakeholders which includes doctors, nurses, and hospital administrators. Here we need planning while deployment which includes medical staff training programs and phase-wise implementation for various hospital departments. Here feedback is very important as it helps us keep on improving our model.

Suitability to our System

i) Pros

- Risk Management Focus
  This is critical for patient data security, and it protects sensitive patient information. It also makes sure we can identify healthcare compliance issues.

- Strong Documentation
  Supports various healthcare field requirements and provides comprehensive requirement tracking.

- Quality Control
  This model performs rigorous testing at each spiral with continuous validation with medical standards. It also ensures regular security assessments.

ii) Cons

- Time Consuming
  This model has longer development cycles and has a lot of extended planning phases. As it has a lot of initial procedures and documentations there will be a delay in deployment initially.

- Resource Intensive
  Not only is this model time consuming, it also results in higher development costs and has a very complex project management.

Final Verdict

The Spiral Model is suitable for Oracle Health System compared to that of the waterfall or incremental model as its risk driven approach matches healthcare's primary concern for patient and data security. With the spiral system it can easily meet regulatory requirements like HIPAA due to its systematic risk assessment and documentation approach. While the Spiral model does require more time & resources, these investments are justified for Oracle Health where system failures could impact patient safety and care delivery.

Summary of all the Models

SDLC MODEL	FLEXIBILITY	RISK MANAGEMENT	TIME MANAGEMENT	COST	SUITABILITY
Waterfall	Low	Low	Medium	Medium	Not Suitable
Incremental	High	Medium	High	Medium	Can be suitable
Spiral	Medium	High	Low	High	Highly Suitable

Requirements Engineering for Oracle Health

Requirements Engineering is fundamental for Oracle Health's System Development. As it is a critical healthcare platform, it must consistently deliver high performance, reliability, security, and seamless user experience while also maintaining strict compliance with healthcare regulations.

Functional Requirements

i) User Authentication & Access Control

It should make sure that there is multi-factor authorization for healthcare professionals using HIPAA-compliant protocols. It should also make sure that there is a Role Based Access Control System put in place for different staff levels (doctors, nurses, administrators). If session is left unattended for too long, it should securely log out of the account.

ii) Billing & Insurance Processing

It should integrate with multiple insurance providers, automated claims processing and verification. It should also support various payment methods and financial assistance programs.

iii) Pharmacy Management System
It should have a electronic prescription management, drug inventory control and also should be able to integrate with external pharmacy systems.

iv) Appointment and Resource Management

It should ensure efficient healthcare delivery through smooth patient scheduling, staff rotation planning. It should make sure there is a count of all resources in the hospital or place where the system is implemented.

Non-Functional Requirements

i) Performance Specifications

It should guarantee 99.99% uptime with maximum 3 second response time. And it should support over 10,000 con-current users. It should also make sure there's no data loss.

ii) System Reliability

The system should maintain constant backups, can recover quickly when it encounters any issues. It should also make sure even in case of crashing it should not conclude in any data loss.

iii) User Friendly Design

The interface should be fit for modern smartphone apps. It should be able to work in all devices from phones to PC or a tablet during medical rounds. The experience should be the same for all types of devices and it should not vary from device to device.

iv) Maintainability & Upgradability

Must include continuous updates and the system should be maintained at equal intervals of time even if the system doesn't have any errors.

Requirements Validation Strategy

i) Initial Documentation Review

The process starts with comprehensive documentation reviews involving multiple healthcare stakeholders and other people involved in the project. They collaborate to examine all requirements documentations. These sessions focus on clinical workflows, security protocols, regulatory requirements are accurately documented and fully understood. This makes sure to identify any gaps or inconsistencies in the specifications before development work begins.

ii) Prototype testing and Feedback

It develops interactive prototypes for critical system functions, particularly focusing on patient record management, prescription systems, and emergency response protocols. This validation helps identify usability issues and workflow inconsistencies early in the development cycle.

iii) Compliance and Integration Testing

Oracle Health conducts thorough testing to ensure that the system doesn't break any healthcare laws, and it should interact with all medical equipment properly. This includes checking security measures and make sure patient data stays protected while there is an information flow between the hospital systems.

Challenges in Requirements Validation

i) Technical and System Integration

One of our main challenges is in managing complex integrations across diverse medical systems while also maintaining optimal performance.

ii) Regulatory Compliance and Patient Safety

The system must maintain strict data protection standards while also being able to adapt to various healthcare regulations.

iii) Resource Management and Timeline Constraints

Limited availability of healthcare professionals for testing, combined with pressure of rapid deployment creates significant time constraint issues.

Conclusion

Based on the analysis from the paper, we can conclude that the Spiral Model works the best for Oracle Health and is highly effective for a healthcare system development model. The spiral models emphasis on risk management and stakeholder involvement makes it particularly suitable for healthcare systems.

Citations

1. Geeks for geeks Software Development Models :

https://www.geeksforgeeks.org/top-8-software-development-models-used-in-industry/

2. Oracle Health :

https://www.oracle.com/in/health/

3. Draw.io

https://app.diagrams.net/