Summer Internship Report

At

Karkinos Healthcare Pvt. Ltd, Bangalore

(April 18th – June 18th, 2022)

Internship report

By

**Manish Bajaj**

PG/21/145

PGDM (Hospital and Health Management),

2021-2023

International Institute of Health Management Research,

New Delhi

# Acknowledgement

I received a fantastic opportunity for learning and professional growth during my internship with Karkinos Healthcare Pvt. Ltd. in Bangalore. As a result, I view myself as a really fortunate person who was given the chance to join this Karkinos team. Having the opportunity to meet so many lovely people and professionals who guided me through this internship time makes me grateful as well.

The **Vice President (Product), Dr. Sandipan De**, who despite being incredibly busy with his tasks and obligations, took time out to bear, guide, and keep me on the right path by giving me responsibilities and letting me do them, deserves my sincere gratitude and special thanks.

I want to give **Chief Product Officer Manish Sharma** my sincere appreciation for helping me out whenever I needed it, as well as for providing the required guidance and counsel at every turn and setting up all the facilities to make life easier. I've decided to express my gratitude for his contribution right now. He is the person I have encountered at this time who has been the most uplifting and understanding.

I would like to express my sincere appreciation to the entire Karkinos product team for their invaluable guidance and assistance in helping me grasp product management from the outset, both of which were crucial to my studies both theoretically and practically.

I see this chance as a significant turning point in my professional progress. In order to achieve my intended career goals, I will make every effort to utilize newly acquired skills and information to the fullest extent feasible and to keep working to enhance them. I look forward to working with every one of you in the future.

**Manish Bajaj**

KARKINOS

Date: 21-Jun-2022

**Internship Completion Certificate**

**To Whom It May Concern**

This is to certify that **Manish Bajaj,** has worked as **"Volunteer Intern"** with Karkinos Healthcare Private Limited and has successfully completed the internship under the guidance of Sandipan De.

Internship Duration: 18th April-2022 to 17th June-2022.

We wish all the best.

Karkinos Healthcare Pvt. Ltd.

**Pooja Sharma**
**Vice President-HR**

# FEEDBACK FORM

**(Organization Supervisor)**

**Name of the Student:** MANISH BAJAJ

**Summer Internship Institution:** KARKINOS HEALTHCARE PVT. LTD.

**Area of Summer Internship:** PRODUCT MANAGEMENT INTERN

**Attendance:** 100%

**Objectives met:** Yes

**Deliverables:** He was an integral part of the product team, from taking part in meetings to helping draft PRDs, User Stories & Journey, Wireframes & research work. He understood the basics of Software Product Management and was able to put forward valuable suggestions. Delivered Software Product Management Toolkit

**Strengths:** Problem Solver, hardworker, punctual, creative

**Suggestions for Improvement:**
All the objectives we set out to achieve have been met.

**Signature of the Officer-in-Charge (Internship)**

**Date:** 15-06-2022
**Place:** BANGALORE

# Certificate of Approval

The Summer Internship Project titled **"Efficient product management for developing robust healthcare softwares"** at **"Karkinos Healthcare Pvt. Ltd."** is hereby approved as a certified study in management carried out and presented in a manner satisfactorily to warrant its acceptance as a prerequisite for the award of **Post Graduate Diploma in Health and Hospital Management** for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein but approve the report only for the purpose it is submitted.

**Ms. Divya Agarwal**
**Associate Dean- Admissions & Accreditation's**
**IHMR- Delhi**

# FEEDBACK FORM

(IIHMR MENTOR)

**Name of the Student:** MANISH BAJAJ

**Summer Internship Institution:** KARKINOS HEALTHCARE PVT. LTD.

**Area of Summer Internship:** Product Management Intern

**Attendance:** 100 %

**Objectives met:** YES

**Deliverables:** → Weekly progress updation
→ Methodology finalization

**Strengths:** Hardworking

**Suggestions for Improvement:** Need to work on the communication skills, more clarity of speech.

**Signature of the Officer-in-Charge (Internship)**

**Date:** July 1, 2022
**Place:** DELHI

# Table of Content

# KARKINOS HEALTHCARE PVT. LTD, BANGALORE


## Organization Objective

### Vision-

Powered by 4D's:-

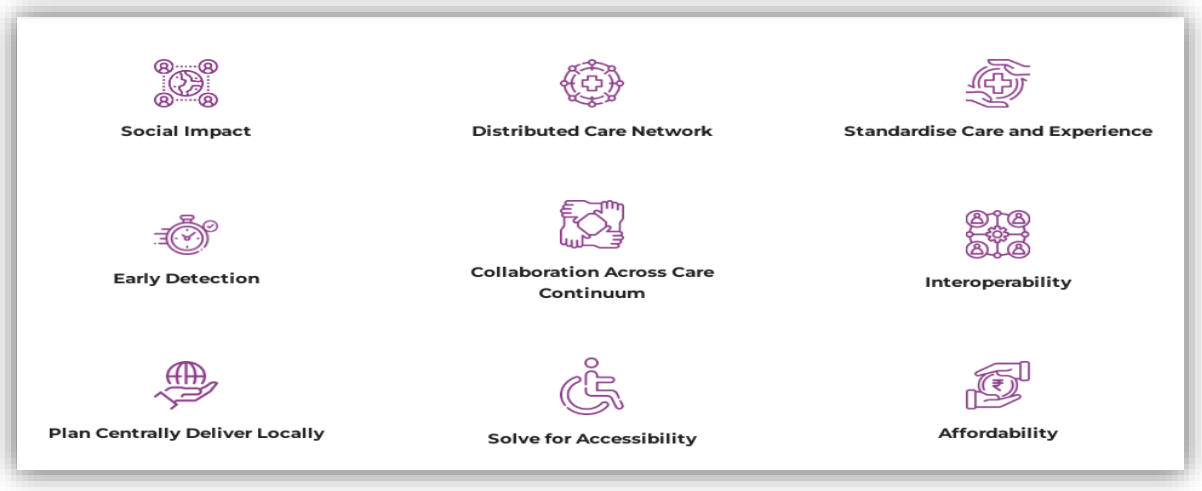
- "**D**etection & **D**iagnosis Establishment of participatory systems and near homecare research on genomics as a foundation approach for prevention, innovation and game-based outreach approach for early diagnosis and wellness".
- "**D**eliver managed healthcare 2 million+ patient hours saved annually".

- "**D**ata and research contribute towards Atmanirbhar Bharat through drug discovery research and treatment innovation large scale screening and longitudinal data to build robust AI/ML analytics and predictive models".

## Mission-

"Our mission is to ensure that no person is denied care either due to access or affordability. Since we are diagnosing cancer at an early stage, the cost can be brought down considerably. Almost 60 per cent of the cost associated with cancer is for indirect expenses. We navigate the care for patients with the existing Karkinos centres and help to carry out some of the treatments such as general surgeries, chemotherapy at locations close to the patient's home, and refer them for radiotherapy and complex procedures to super speciality cancer hospitals"

## Values-




**Social Impact-** Reduce travel time from 8-10 to 3-5 hours; 30-40% reduced cost of care Surveillance & early detection. Early detection, Better reach out & care, improved outcomes.

**Distributed Care Network-** Reversing current focus from treatment to early detection is possible with a Distributed Care Network.

**Standardise Care and Experience-** Standardised and Patient centric pathways, improve quality in delivery of care and patient outcomes.

**Early Detection-** Early Detection Enables Stage Shift in Cancer Incidence

**Collaboration Action Care Continuum-** Patients Centric approach that enables Care Collaboration across the Patient Care Continuum.

**Interoperability-** Enabling the Flow of Patient Information by adhering to the interoperability standards identified by the NDHM

**Plan Centrally Deliver Locally-** Command Centre to enable distributed deliver care delivery by a centralized knowledge architecture.

**Solve by Accessibility-** In a Patient Centric Model, Karkinos enables access to care near the patient's home driven by geo-tagging of care partners.

**Affordability-** A cancer care network enabling early detection with assured, affordable and quality focused treatment pathways.

## About Karkinos

**Karkinos Healthcare Pvt. Ltd**, a purpose driven technology-led oncology platform, is focused on designing and delivering bespoke solutions for cancer care. The company, led by a blend of globally acclaimed medical professionals and technologists, is on a mission to create 'cancer centers without walls with a primary aim of addressing the accessibility or affordability gaps in cancer care.

Karkinos ('Kark' means cancer in Hindi and 'nos' means elimination) intends to set up 70 community care centres across the State in a year to offer distributed care. It plans to expand throughout the country's health system using a technology platform.

Karkinos is concentrated on addressing clinical demands through a dispersed cancer care network that is digitally enabled and will bring high-quality care closer to patients. The cornerstone of Karkinos health is the democratisation of cancer care through collaborative

efforts with current medical professionals, researchers, and technologists. Karkinos Healthcare is developing a medical facility for the treatment of complex cancers, an open standards-based technology platform that coordinates the cancer care continuum, and a research facility that makes use of technologies like genomics, synthetic biology, sensors, and AI to analyse data and advance the development of reasonably priced cancer interventions. The core of the technology platform is an openEHR-based Clinical Data Repository. is hereby acknowledged as a management research that has been completed and presented.

The clinical operations of the first such community centre has already started at Kothamangalam in Ernakulam district a few months back and other centres were opened recently at Chottanikkara, Thodupuzha, Munnar both in Ernakulam and Idukki.

It is estimated that there are 2.25 million cases of cancer in India every year, which doubles every 10 years. Three quarters of these cancers are detected in the late stages and mortality rates are devastatingly high because of lack of access to standardized cancer care. Whilst Indians are at the forefront of medical research in the West, India as a country is a laggard in researching and curing the condition.

Recently, Mayo Clinic of the U.S. has invested in a minority stake in the company, subject to certain conditions precedent.

Mayo Clinic will also nominate a member on the Board of Directors of Karkinos as part of this agreement besides transferring technical know-how. The size of investment has not been disclosed.

The distributed cancer care Network model is being developed in India by Karkinos Healthcare, which has as investors Ratan Tata, Venu Srinivasan, Kris Gopalakrishnan, Ronnie Screwvala, Vijay Shekar Sharma, and Bhavish Agarwal.

A minority share in Karkinos is held by Reliance Digital Health, a subsidiary of Reliance Industries, and Rakuten Medical, a global biotechnology business in the clinical stage. The Tata Group has invested 110 crore in Karkinos. Endiya Partners, a venture capital firm, also owns a stake in the business.

With an end-to-end technology platform that coordinates the cancer care continuum, a medical centre for treating complex cancers, and a research centre that uses technologies like genomics, synthetic biology, sensors, and AI to analyze data and create accessible cancer interventions, Karkinos Healthcare intends to address the problems with cancer care. The cornerstone of Karkinos health is the democratisation of cancer treatment through active collaboration with current healthcare professionals, researchers, and technologies.

Karkinos Healthcare has set up a command centre to support the patient to undergo various components of treatment and an Advanced Cancer Diagnostic Laboratory in Kochi to guide the patient for treatment.

Cancer is the second most common cause for deaths in the country and 1.3 million new cases occur every year. Around 3.6 million are living with cancer and require medical help. In Kerala, as many as 66,000 new cases occur every year and around two lakh people live with the disease.

## Observation Learning at Karkinos Healthcare

At Karkinos healthcare I was appointed as a Product management intern for a period of two months, there I learned and observed many important things which was required for my internship. I was in the product team of Karkinos, in which there is product analyst team, creative team and the most important developer team.

**Product Team:** In product team, there are a lot of people from top to down and they all worked together to build the best possible product. Here I have learned new things about Product management starting from basics of it.

In the first one month I learned about the basics of Product management, which is necessary to understand the things which is essential for product delivery.

First I learned about the Product management, in simple language it is an organizational role that directs each stage of a product's lifecycle, from development to positioning and price, by placing the customer and the product first.

Then I used to learn important steps required to make a product, starting from the **question, hypothesis and research** required to make a product, first the question is Why we need to have to that product/ feature or the service, for that we have to understand the need behind the requirement so that we can know what kind of requirements the customer has. A hypothesis is a claim made based on incomplete information about a situation that needs to be validated in order to be established as true or untrue to the extent necessary for the team to move forward with their study and identify the best solution to the issue at hand.

After getting the requirement or the research hypothesis, I learnt about **Market research**, in which its essential to know the requirement of market, as well as the trend & taste of customer who are going to use the product/ service.

Then I worked on **User story**, which is very important for any feature as it is a casual, everyday description of a software system's features. It may be recorded on index cards or digitally in product management software, and it is always written from the viewpoint of an end user or user of a system.

On the similar time, I also learned about **User testing** which is a procedure used to evaluate the user experience and features of a website, programme, mobile application, or service. Particularly, actual users who conduct the user testing procedure under actual conditions are involved in this examination of the user testing process.

Then I got the opportunity to know about **Product lifecycle**, it is a type of product roadmap which is a strategy for how a solution or product will change over time. Roadmaps are used by product owners to specify future product functionality and the timing of the release of new features. When utilized in agile development, a roadmap gives the team's daily work critical context and should be adaptable to changes in the market. A product roadmap is a source of truth that is used by all parties to describe a product's long-term goals, priorities, and progress. It's a strategy that unites the company around the project's short- and long-term goals and how they will be attained.

**Software development lifecycle,** process is typically drawn out and tiresome. However, product managers and system analysts can make better use of software development life cycles to define, design, build, test, and ultimately deploy information systems or software products.

It outlines the various steps required to take a product from its initial conception or idea all the way through deployment and subsequent maintenance. Software lifecycle has some phases: - Requirements analysis, Software Design, Software Development, Testing, Deployment, and Maintenance.

A **Model for Agile**, the goal of software product management is to create software as quickly, cheaply, and at the highest possible quality. It also contains comprehensive instructions on how to create, enhance, and manage the software system. Agile also refers to the "capacity to move swiftly and readily" and quickly adapt to change, which is a key component in agile software development. The main reason why the Agile is mostly used in software development because there is daily interaction between the consumer and the product owner, so if the customer has any input or wants to change a feature, it may be accommodated in the current release of the product.

**Creative Team**: From the last week of first month of my internship, I worked with designer team. The creative Designers are responsible for giving priority to their team's projects. They should be aware of project timings and should manage their own planning based on it. Designers are also responsible for giving visibility of all their projects to their own squads.

The Design lead helps designers to be more efficient and should be aware of the designers load in order to prevent load issues and optimize their effort.

**User Experience (UX)**: A set of tasks that enable effective and enjoyable use of a product/software.

**User Interface (UI)**: A subset of UX. It's the design, feel and interactivity of product/software.

**PRD**: It is a Product Requirement Document is used to create an artefact in the product development process to inform the development and testing teams of the features that must be included in a product release. As a guide for following documents in the release process, the PRD will cover everything that a release must include in order to be considered complete. Product Requirement Document includes Objective and goals, features, user story, user journey, process flow diagram, design flow, System and environment requirement and last is Assumptions, constraints and dependencies.

**Process Flow Diagram:** A process flow diagram is a picture of the separate steps of a process in sequential order also used to demonstrate the functionality/feature of a software from the perspective of a single end user. It is a general tool that may be used for many different things and used to define different processes, like the service process.

**Swimlane Diagram:** A Swimlane Diagram, is similar to PFD's, only that it involves two or more than two types of process/ feature / department and establishes a relationship between them in entire flow of a particular feature/functionality. In comparison to a standard flowchart, it clarifies roles better. The process of eliminating delays and inefficiencies can be speed up when attempting to optimize processes by identifying which department is in charge of what.

**User Journey:** It is a representation of a person's relationships and feelings about a good or service over time and through various media. It was depicted as a chronology of all interactions that have ever taken place between a user and a product. A user journey map assists a product team in answering hypothetical queries. A user journey map might also be useful if the business monitors quantitative KPIs.

**Double Diamond- Design Thinking:** In the double diamond is used to teach the lesson that without a clear knowledge of the design strategy and characteristics, any design thinking process is meaningless. Design thinking's primary goal is to provide direction for using this tactic in practical situations. The two diamonds reflect a process of engaging in more in-depth or broad analysis (divergent thinking), followed by targeted action (convergent thinking). The four steps of the design process are suggested to be Discover, Define, Develop, and Deliver.

**Wireframes:** A wireframe is a schematic or blueprint that can be used to facilitate communication between you, your programmers, and your designers regarding the organizational layout of the software or website you're developing. Wireframes can range from Low (hand drawn sketches, less detailed concept) to High (Highly Detailed Render of prototypes) Fidelity. Some applications for making wireframes: Figma, Adobe XD, Balsamiq, etc.

**Developer Team:** In the second week of second month, I used to get interaction with the developer team by which I got to know about the basic things in the developer team, there I didn't get any work but they shared knowledge and other such things which was insightful.

There are many people working in the developer team from frontend to backend. There main work start after from the beginning of the product design by the product team when they got the requirement about any product/ feature from the customer to ask from the frontend team, as if they make a changes in the present feature, will it hamper the flow of the software

Same as after preparing of the designs of the required feature by the creative team, they give the designs to the Backend team to prepare/ or enroll the prepared designs in the product. These all works are done by the backend team by constant communication between the product team, creative team, frontend team and the backend team.

**Frontend Team:** The area of a website that users interact with is called the front-end. Your computer's browser uses a combination of HTML, CSS, and JavaScript to control everything you see while browsing the Internet, including the fonts, colours, drop-down menus, and sliders.

**Backend team:** A server, an application, and a database make up a website's backend team. The technology that drives the many components that collectively allow the user-facing portion of the website to even exist is created and maintained by a back-end developer.

**Testing:** Before testing for mass production, there is one step left. It comprises of testing, quality assurance, and quality control. Despite the fact that, depending on the size of the project, they are technically distinct stages in the development process, stakeholders frequently combine them because they all aim to produce a high-quality product.

**Maintenance:** After a software product has been delivered to the customer, it can still be modified through a process called software maintenance. The primary goal of software maintenance is to maintain and alter software programmes once they are delivered in order to fix bugs and boost performance.

**Minimum Viable Product:** The term "minimum viable product," or "MVP," refers to a product with just enough features to draw in early adopters and validate a product concept. In sectors like software, the MVP can assist the product team in gathering customer input as soon as possible so that they can iterate and enhance the product.

Other than these, I also got an opportunity to work on **Custom field**, in which I worked on the all the things together, such as PRD, User story, user journey, wireframes and also process flow diagram. A custom field is the addition of the new field/ feature in the already establish product. It is usually done to for the indulging of more information which may require for the betterment of customer, or may get some benefit by giving some additional information along with the compulsory or mandatory fields.

One more opportunity got to learn about how to **Prioritizing product backlogs**, it is required to maintain or prioritize the backlogs for each organization because it will help in getting more revenue for the organization also if the organization don't do this on a given time it will hamper the progress of the product. It's the nightmare of any Product team is having a huge unorganized backlog with no resolution. So first I will discuss about how the backlogs are built as it is usually been done due to again and again delay in the task in current sprint which results in to include the tasks in next sprint as it comes in the sprint backlog list.

As there are many methods to prioritize the backlogs but as per my learning I will discuss here the best possible methods of prioritizing-

**Opportunity Scoring-** By identifying the aspects that customers deem critical yet underdeveloped or otherwise disappointing, it is possible to prioritize feature development. Opportunity score can be compared to an examination of importance vs. satisfaction. We use customers' ratings of the significance of various product features and their satisfaction levels to conduct opportunity scoring. Our opportunities are those characteristics that receive high significance ratings but low satisfaction ratings. In other words, these features show great promise for the money and time we will need to devote to developing them.

**MoSCow Method-** In Agile teams, mostly the Moscow Method is used because of its simplicity and its ease of use. Also, MoSCoW allows teams to determine the relative importance of Backlog items which means any Backlog item required for a product to function. Should stands for items that have high value to the customer. Could Backlog items are small fixes and features. Would items have the lowest importance and can be added to a future Backlog if the team doesn't have time.

**Value vs Complexity Matrix-** This model is one of that many product managers regularly use to prioritize product roadmap initiatives. It offers a standard approach to decision-making. It is useful for being more strategic about what initiatives to give attention to, especially when you have limited time or resources. Value vs. complexity is a matrix featuring four quadrants, which form categories into which you place initiatives. The idea is to identify low-hanging fruits you can start with.

## Conclusive Finding

Overall I learned many things during my 2 months of my tenure at Karkinos, first about the product team in which I learned about basics of product management, product lifecycle, Software Development Lifecycle and models required to make a product. Then I got to learn about the creative team in which I learned about the user interface (UI) and user experience (UX), after that I worked on PRD's and its components and wireframes. Then when I was having an experience of these, I worked on custom fields, which covers all the things from requirement analysis to PRD to wireframes. At the end I learned about the methods of prioritizing product backlogs and was involved in taking presentations on the product management and the Deep Medicine by author Eric Topol.

# Project Report

# “Efficient product management for developing robust healthcare software’s”


## Abstract

This narrative review was conducted to understand the basics of product management in developing efficient healthcare software’s and applications. A software is a component of the product which a higher order thing. In the fast paced ever evolving healthcare scenario development of such software’s is usually plagued by various hurdles. The healthcare information technology field is little behind on progress in comparison to other domains. Most of the healthcare software’s ranging from standalone services to complete EHRs/EMRs are not user friendly leading to physician burnouts, mistreatment and misdiagnosis. The current trends of product management approach attracts more than needed alteration of end product resulting in low quality healthcare software’s. It becomes crucial for aspiring product mangers to understand the basics of Software Development Lifecycle (SDLC) and it various management models. It’s important to choose an iterative approach towards software development while taking into consideration the viewpoint of end users and customers to develop and deploy effective and efficient software’s and applications which are enjoyable to use. Periodic backlog grooming along with employment of automation makes the overall process easier while reducing burden on the product team. Application of smart and flexible thinking governed by good product management and general management principles becomes crucial in developing such software’s.



## Keywords

Product management, Roadmap, Software Development lifecycle (SDLC), SDLC models, UI/UX, Test Driven Development, Backlogs

## Introduction

A product is a delivery that gives its customers value and an enjoyable experience. It may consist of a variety of supplied systems, solutions, materials, and services. In the context of healthcare software's and applications, it's important to understand that a product is a higher order thing and a software is a component of the higher order. A software can be used as a service and as a product. Software as a product will require purchase of licensing under which various solutions and services can be availed offered by the software. A service is an ephemeral, non-tangible product that does not involve a transfer of ownership and is the outcome of at least one activity conducted at the interface between the supplier and the client co-creating value. A solution is a customized product made for a particular business or consumer requirement out of several products, processes, and resources. (1)

Usually there is a rift between delivery and marketing teams with both exerting pressure on each other to meet their respective deadlines for the release. This leads to dysfunction within the team. Hence Product management, which encompasses the full lifecycle and thus ensures both a technical and business perspective, is a critical success factor for successful product launch and sustainability. The need for such a product management is often felt more in technology companies that have a laissez-faire culture. (2) Such a free hand, no action culture does not usually work out well in the healthcare information technology field as there is massive competition and developments happening at a fast pace. Therefore, to achieve greater flexibility and better satisfy customer expectations, whether it be a Business to Business (B2B) or Business to Consumer (B2C) transaction, an incremental software development technique regulated by excellent product management principles is typically implemented. (3)

It's important for a product manager to understand the Software Development Life-Cycle (SDLC) and the various management models employed to ensure value orientation and consistency across the healthcare software releases. (4)

## Software Development Lifecycle (SDLC)

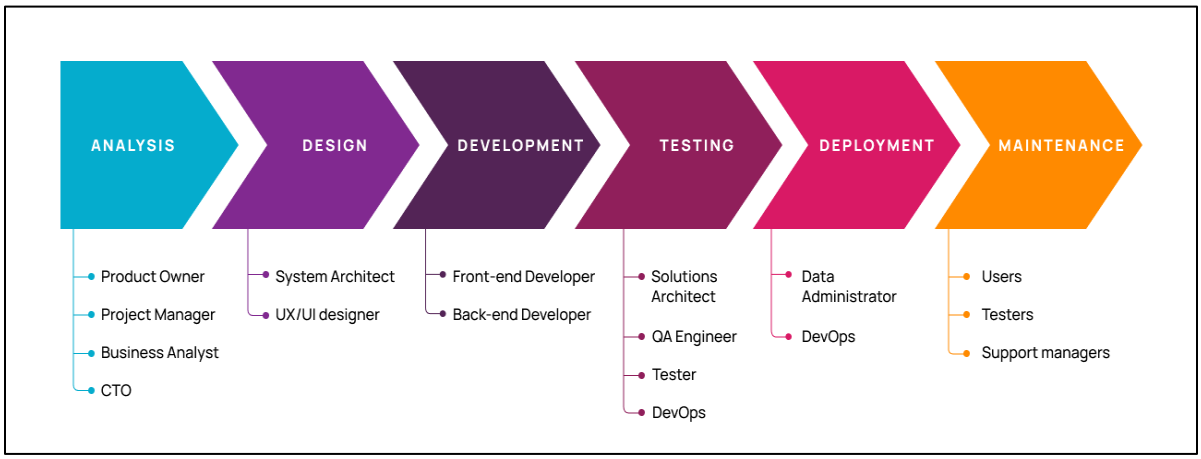



Figure 1.Stages of Software Development Lifecycle (SDLC) governed by various members Healthcare IT Team

Requirements analysis/requirements engineering, the first step in the SDLC, is essential for comprehending the software/product roadmap. Product managers utilize these roadmaps to describe future functions and the timing of the introduction of new features since they provide a plan of action. In a sense an appropriate market analysis of healthcare needs and concerns along with understanding how competitors are performing will shed light on vision, direction, priorities, and progress of the healthcare software/application over time. (5) The next aspect of design is extremely important as it dictates the core User Interface (UI) and User Experience (UX) which regulates the effective and enjoyable use of the software.(6) This becomes crucial in the healthcare setups may it be for separate services like telemedicine, health chat bots, health wearables using IOT (smart watch, fit-bits, at home ECG monitoring patches, medical devices), E prescriptions, online appointment scheduling within standalone license based softwares/ products to complete Electronic Health Records (EHRs) and Electronic Medical Records (EMRs) encompassing all these services along with vast multitude of more services such as Clinical Decision Support System(CDSS),etc. required by hospitals.(7) In the current day and scenario such softwares especially the EHRs/EMRs are very clunky to handle resulting in physician burnout leading to mismanagement and misdiagnosis of patients. Hence, the design step needs to ensure the vision of creating a user friendly and efficient experience while navigating such softwares. (8)

The development phase is what brings the design into life with the backend developers using appropriate coding to develop the tech stack while enabling frontend developers to enforce Application Programming Interface (APIs), plugins, version updates and automation without disturbing the backend. (9) A tech stack sometimes also referred to as solutions stack is nothing but several layers of code which forms the foundation of any software and it's important to remember that it's quite hard to rip off a layer from this foundation as it can result in bugs and glitches in the software. A well-built tech stack with a good frontend interoperability offers smooth communication between different software's and components. (10,11) From the perspective of the legal product requirements, it's important to remember that the platform being built should concede with the Data Protection and IT Laws governing each country or region as healthcare data often contains sensitive patient information. (12)

Testing is a important step in accomplishing and evaluating the quality of a software product. Testing is performed at different levels throughout the life cycle of a software product. (13)

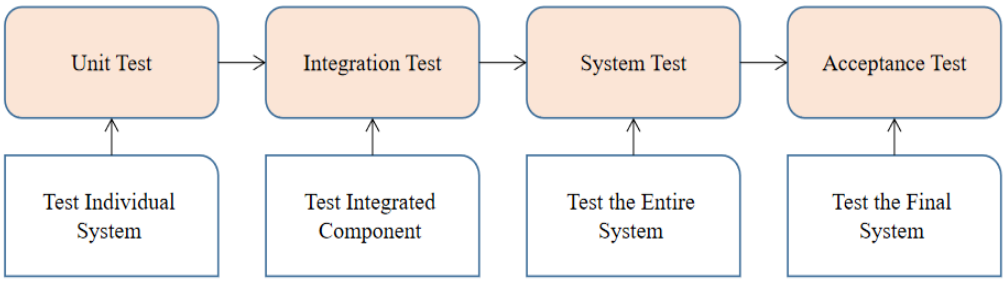



Figure 2. Levels of Testing

During and after the development phase the software undergoes testing where it's scrutinized to various regression tactics to see how it performs and behaves in a controlled and live environment. These testing methods include white box (structural) and black box (functional) testing. Traditionally testing was done after development, but in recent years the applicability of such traditional methods has resulted in loss and damage. Hence, a test-driven development (TDD) is often followed nowadays which overcomes such inefficiencies. (14)

Once it passes all the required check-lists either a Minimum Viable Product (MVP) or complete software with almost all features is deployed.(15) Since the Healthcare field is constantly evolving especially after the inception of Covid, these healthcare SDLCs usually

employ the use of MVPs as this iterative approach ensures there is no feature creep and the software remains light and easy to use with features being added later depending on customer and market requirements. Hence several consistent versions will be launched periodically with additional relevant features being added over time. (16) However this iterative approach also attracts several requests for features every now and then if the roadmap is not well planned.(17) Considering all above scenarios are well met, the final phase of maintenance is an everlasting phase until the software goes out of market. This phase ensures that the software stays on par with developments of the healthcare field by launching periodic version updates and feature additions to increase the longevity of the healthcare software/service/application.(18) The maintenance phase has four stages as shown in Figure 3, which ensure the longevity and efficiency of the software. (19)

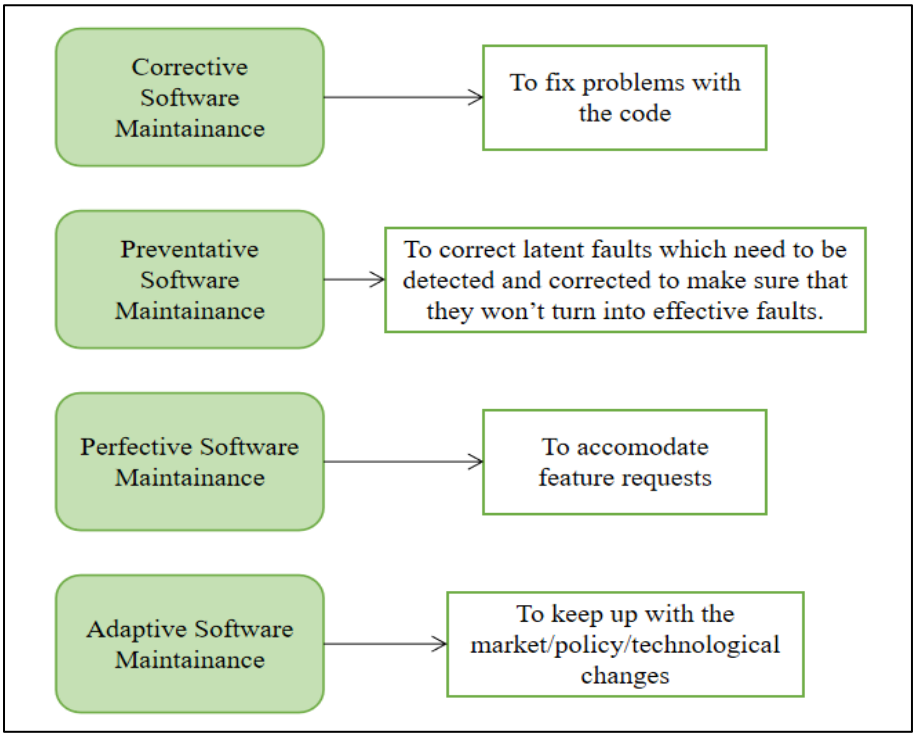



Figure 3. Four phases of Software Maintenance

## SDLC Management Models:

There are various management models and the most used ones are: waterfall, spiral, Evolutionary prototyping, rational unified process (RUP), agile, v-shaped management

models. The ability to understand which model is ideal for what situations is extremely important. The degree of difficulty of the programme that needs to be created is also important. Agile approaches are typically appropriate for small projects. A more thorough approach using a combination of various lifecycle models may be necessary for larger projects. (20) Healthcare software's usually work best when developed using an iterative model. (21)

The Waterfall model shown in figure 4, also known as the grandfather of all lifecycle models is an ideal management model when the domain is well understood. (22) This ideal situation is not seen in the healthcare field as its ever evolving, hence this model falls short in this context. Advantages of waterfall model: early detection of errors. Disadvantages include lack of flexibility and its less than ideal for most real world applications.(23)

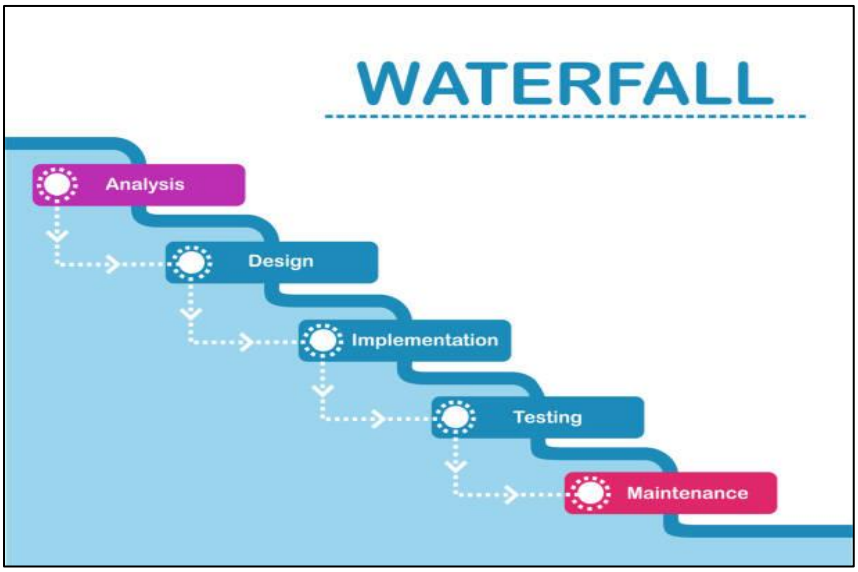



Figure 4. Waterfall Model

Creator: Sylfide | Credit: Getty Images/iStockphoto

Spiral Model shown in figure 5, is an incremental risk oriented life cycle model. This Model replicates a project's steps by beginning with modest objectives and spiraling outward in ever-wider directions (called rounds). The spiral's rounds each represent a project, and each cycle may use a conventional software development process like modified waterfall. (24) Each round includes a risk analysis. Early phases are more likely to uncover fundamental issues with the project or process, leading to easier changes. By identifying and mitigating significant hazards, the project's overall risk is reduced. Advantages of spiral model: Risk reduction, functionality

can be added in later phases and software can be produced early with core features. Disadvantages are that this model requires specific expertise, its complex and is highly dependent on risk analysis. (25)

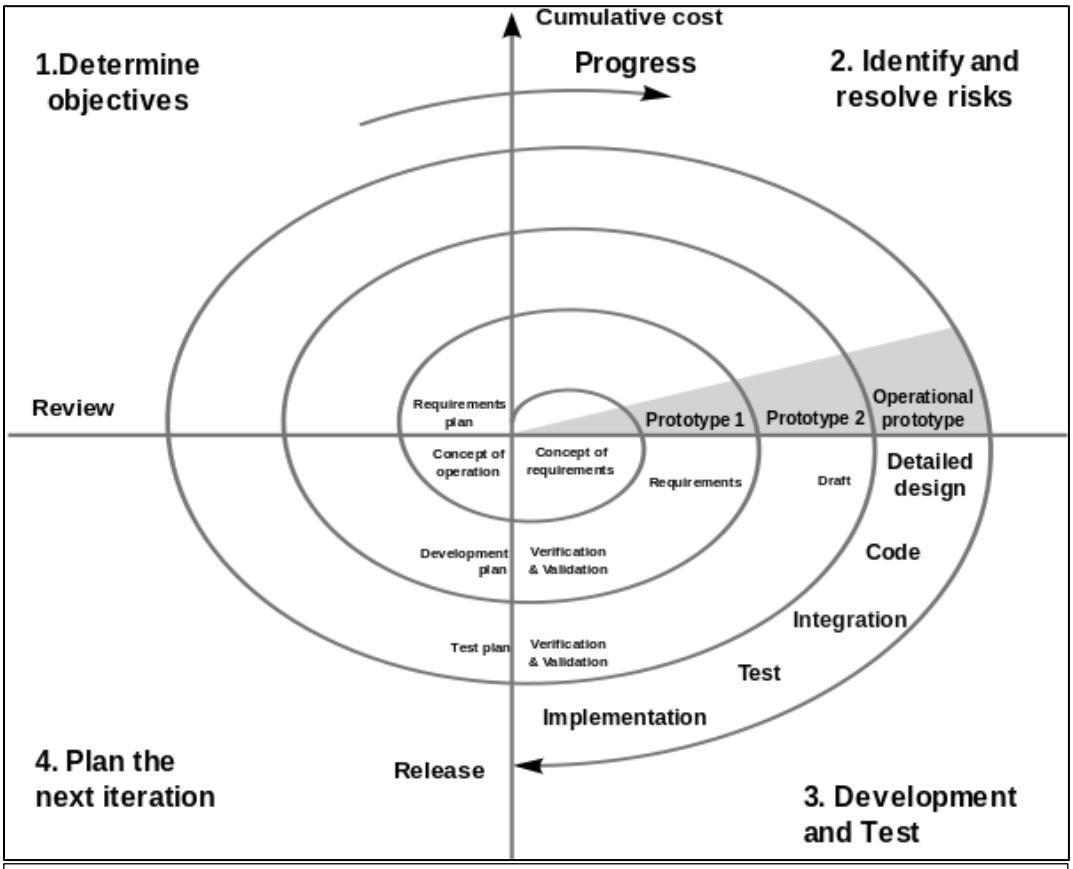



Figure 5. Spiral Model

Source: https://www.pngkey.com/detail/u2q8q8o0w7e6a9w7_spiral-model/

Evolutionary prototyping model shown in figure 6 offers prototypes which are continuously refined and rebuilt until a complete version with all features is achieved. In this model the developers start by developing parts of the system that they understand instead of the whole system. Advantages of evolutionary prototyping include immediate feedback, Its an ideal model when all requirements are not well understood. Disadvantages include that this model is difficult to plan, and it entertains excuses to cut and fix features as per customers resulting in a low quality software. (26)

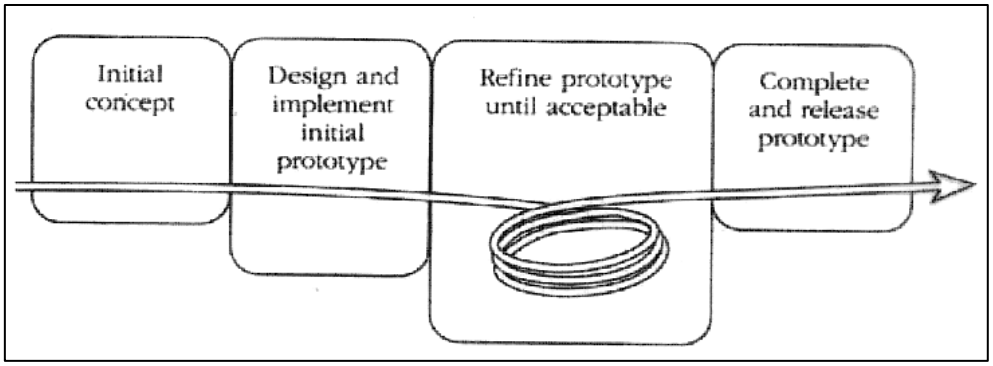



Figure 6. Evolutionary Prototyping

Rational unified process (RUP) also referred to as Unified Software Process (USP) shown in figure 7, is a process product which is an iterative model with four phases of inception, elaboration, construction and transition. Rational Unified Process is designed and documented using the Unified Modeling Language (UML). (27)

Advantages of RUP/USP: provides a very coherent backbone to the process, it has a well-defined and regular structure, uses an object-oriented approach for its description, reuse of code is efficient and fast resulting in quick product delivery. Disadvantages include requirement of high expertise, ever changing advancements pose a problem to reuse of components leading to an undisciplined format of software development. (28)

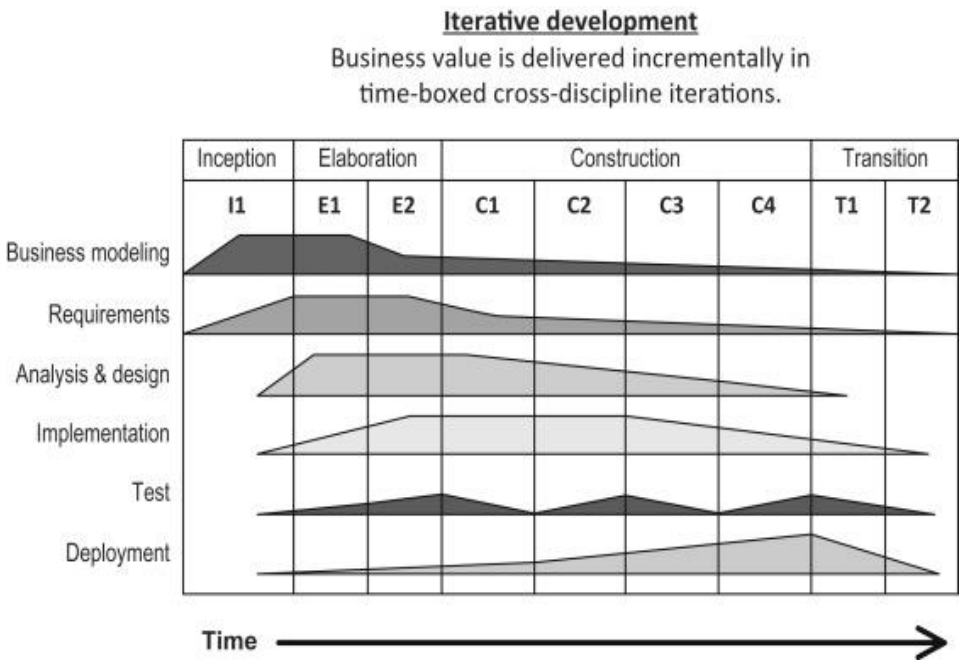



Figure 7. Rational Unified Process (RUP)
Source: https://www.sciencedirect.com/topics/computer-science/rational-unified-process

Agile management shown in figure 8, is one most used management models for creating healthcare software's and applications. It's based on a highly iterative and incremental development and employs TDD like other iterative models discussed above. (29) The major difference of the agile model over other iterative models is that it offers rapid customer satisfaction with continuous delivery of useful software. Software is rarely perfectly fit for use upon first release, but the best software evolves over time to achieve maximal effectiveness. Physicians are more involved in the agile user testing process of each incremental version which provides better insights in what design works best. (30) Major disadvantages of the agile model include too much customer focused delivery, lack of documentation and inability of product teams to adapt to changes to meet deadlines. (31)

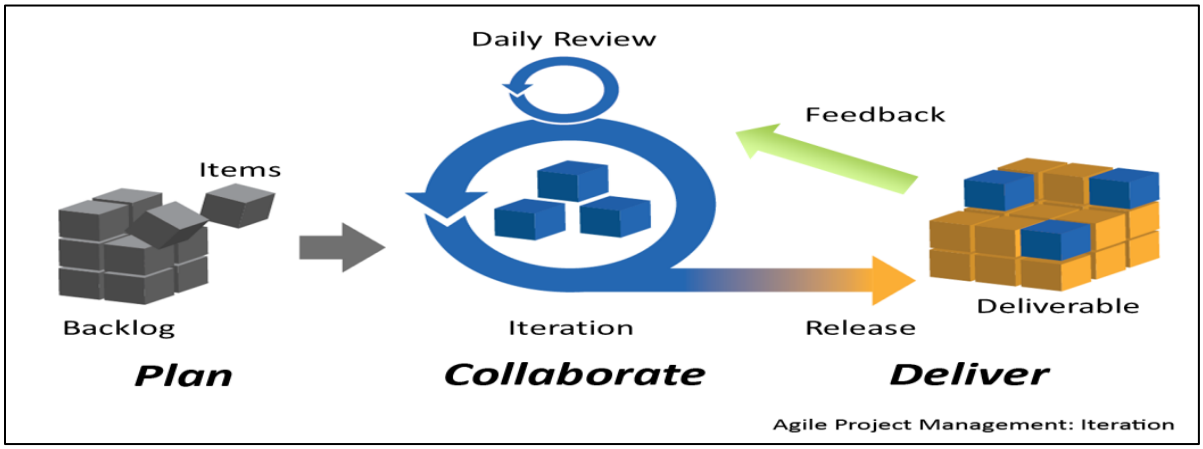



Figure 8. Agile Software development

Source:https://commons.wikimedia.org/wiki/File:Agile_Project_Management_by_Planbox.png

## Product/Software backlogs:

A development team's prioritized list of tasks is called a "product backlog," and it is formed from the roadmap and its requirements. It is a set of tasks needed to complete the final Product. These backlogs usually include bugs, feature requests and enhancements. (32) Based on the items at the top of the queue, developers often create new features, alter existing features, and address bugs. The backlog serves as the principal non-code or software product management asset for many teams. Analysts usually identify and prioritize the backlogs depending upon criticality and value of the backlog. (33) Resolution of backlogs is crucial to ensure effectiveness of software to be delivered. A large unresolved backlog is the nightmare for any product team, hence backlog grooming should be done periodically to understand what is achievable and what is not. Several models are available which help to prioritize and resolve backlogs effectively. Some effective models are MOSCOW, Critical Path Model (CPM), Value vs Effort matrix. These backlog management models when employed right along with regular team meetings can result in effective resolution of backlogs.(34) An ideal backlog resolution is never really complete in a agile environment, however keeping backlogs at the bare minimum should be the aim for any product manager. Breaking the backlogs into smaller manageable chunks, deletion of old backlogs which will never be resolved are some more methods which can make life easy going during the development process.(35)

## Role of automation in development process:

Software automation is an important step in business scaling and software development as it offers substantial cost-saving and helps the IT staff refocus more on strategic activities than administrative functions.(36) Automation of mundane tasks, testing, data cleaning, file organization, important reminders, version updates and controlled access help coordinate activities better in the software development process. (37) It is important to understand that relying on manual and ad hoc solutions should not be an option as it can result in the usage of out of date software versions, loss of old stable versions, data breach, data loss due to network outages and can divert focus of the team on small tasks which are otherwise insignificant. Many IT companies have had huge losses and failures due to not relying on proper automation in the past. (38)

## Conclusion:

Due to the ever evolving scenario of the healthcare field, relying too much on a set path or single technique for software development can never be an option. Several issues hinder the progress of healthcare software development in current day and age. These issues range from product related issues such as gold plating of requirements to technology related issues with too much dependency on the advertised benefits of a previously unused technology and lack of automation. It even extends to application of basic management principles plagued by poor management of employees, self-burden and heroics discouraging team involvement while encouraging unnecessary risk taking. It’s important to keep the expectations grounded in reality. It’s essential for a product manager to fully own the vision and roadmap of the healthcare software product to be developed. The product team should develop a deep understanding of the end users and needs to help create useful product experiences that address those needs. A good ability to push back on software engineering estimates if project is under or over-scoped is essential. The team should have a good knack to understand the technical constraints and should be able to spot performance bugs while synthesizing insights from user research, customer meetings, end user metrics and feedback to prioritize improvements and

new feature ideas. A combination of smart thinking along with application of principles and iterative management techniques mentioned above can help kickstart the software development process in making robust healthcare softwares.

## References:


1. Ebert C, Brinkkemper S. Software product management – An industry evaluation [Internet]. 2022 [cited 20 June 2022]. Available from: https://www.sciencedirect.com/science/article/abs/pii/S0164121214000156
2. Juneja P. The Need for Product Management in Organizations [Internet]. Managementstudyguide.com. 2022 [cited 20 June 2022]. Available from: https://www.managementstudyguide.com/need-for-product-management.htm
3. Greer D, Ruhe G. Software release planning: an evolutionary and iterative approach. Information and Software Technology. 2004;46(4):243-253. Available from: https://www.sciencedirect.com/science/article/abs/pii/S095058490300140X
4. Raza M. The Software Development Lifecycle (SDLC): An Introduction [Internet]. BMC Blogs. 2022. Available from: https://www.bmc.com/blogs/sdlc-software-development-lifecycle/
5. Zhang H, Zhang H, Wang X, Yang Z, Zhao Y. Analysis of Requirements for Developing an mHealth-Based Health Management Platform. JMIR mHealth and uHealth. 2017;5(8):e117. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5561389/
6. Lindberg O. Editorx.com. 2022. Available from: https://www.editorx.com/shaping-design/article/healthcare-ux-design
7. Miller A. Why UX in Healthcare Technology Matters and Where It's Going | 2020. Medium. 2022. Available from: https://uxplanet.org/why-ux-in-healthcare-technology-matters-and-where-its-going-2020-82a2cb09ef55

8. Howe J, Adams K, Hettinger A, Ratwani R. Electronic Health Record Usability Issues and Potential Contribution to Patient Harm. JAMA. 2018;319(12):1276. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5885839/
9. O'Dowd E. What are Benefits of Healthcare Application Backend as a Service?. HIT INFRASTRUCURE. 2022. Available from: https://hitinfrastructure.com/news/what-are-benefits-of-healthcare-application-backend-as-a-service
10. Herman H, Grobbelaar S, Pistorius C. The design and development of technology platforms in a developing country healthcare context from an ecosystem perspective. BMC Medical Informatics and Decision Making. 2020;20(1):55. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7068897/
11. Korkishko I. Healthcare industry trends and development tech stack for 2020. Medium. 2020. Available from: https://medium.com/gobeyond-ai/healthcare-industry-trends-and-development-tech-stack-for-2020-f56e74f06ffa
12. McGraw D, Mandl K. Privacy protections to encourage use of health-relevant digital data in a learning health system. npj Digital Medicine. 2021;4(1). Available from: https://www.nature.com/articles/s41746-020-00362-8
13. Levels of Testing - javatpoint. www.javatpoint.com. 2022. Available from: https://www.javatpoint.com/levels-of-testing
14. Honest N. Role of Testing in Software Development Life Cycle. International Journal of Computer Sciences and Engineering. 2019;7(5):886-889. Available from: https://www.researchgate.net/publication/335809902_Role_of_Testing_in_Software_Development_Life_Cycle
15. Khristich S. How to Build an MVP for Healthcare: a Complete Guide. Tateeda.com. 2022. Available from: https://tateeda.com/blog/how-to-build-an-mvp-for-a-healthcare-product
16. Geevarghese D. Everything You Need to Know about Healthcare MVP Development. Cabotsolutions.com. 2020. Available from: https://www.cabotsolutions.com/everything-need-to-know-about-healthcare-mvp-development
17. Gladchuk V. How to Build an MVP for an IT Healthcare Project. GreenM. 2021. Available from: https://greenm.io/how-to-build-an-mvp-for-an-it-healthcare-project/
18. Al Rababah A, A. Alzahrani A. Software Maintenance Model through the Development Distinct Stages. IJCSNS International Journal of Computer Science and Network Security.

2022;19(2). Available from: https://www.researchgate.net/publication/333176174_Software_Maintenance_Model_through_the_Development_Distinct_Stages

19. The 4 Types of Software Maintenance-What is Software Maintenance. Cpl.thalesgroup.com. 2022. Available from: https://cpl.thalesgroup.com/software-monetization/four-types-of-software-maintenance
20. 11. Sami M. Software Development Life Cycle Models and Methodologies - Mohamed Sami. Mohamed Sami. 2012. Available from: https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/
21. 12. Ivanova D, Kadurin V. A new proposed software development methodology for healthcare industry. THERMOPHYSICAL BASIS OF ENERGY TECHNOLOGIES (TBET 2020). 2021. Available from: https://aip.scitation.org/doi/pdf/10.1063/5.0042261
22. 13. Petersen K, Wohlin C, Baca D. The Waterfall Model in Large-Scale Development. Lecture Notes in Business Information Processing [Internet]. 2009 :386-400. Available from: https://www.researchgate.net/publication/30498645_The_Waterfall_Model_in_Large-Scale_Development
23. 14. Lenz R. A Practical Approach to Process Support in Health Information Systems. Journal of the American Medical Informatics Association. 2002;9(6):571-585. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC349375/
24. Conrad E, Feldman J. Domain 4: Software Development Security . ScienceDirect. 2014. Available from: https://www.sciencedirect.com/topics/computer-science/spiral-model
25. Boehm B. The Spiral Model as a Tool for Evolutionary Acquisition. CROSSTALK The Journal of Defense Software Engineering. 2001;14(5):10. Available from: https://www.researchgate.net/publication/228805054_The_Spiral_Model_as_a_Tool_for_Evolutionary_Acquisition
26. Xiaoshuan Z, Zetian F, Wengui C, Dong T, Jian Z. Applying evolutionary prototyping model in developing FIDSS: An intelligent decision support system for fish disease/health management. Expert Systems with Applications. 2009;36(2):3901-3913. Available from: https://www.researchgate.net/publication/223584799_Applying_evolutionary_prototypin

g_model_in_developing_FIDSS_An_intelligent_decision_support_system_for_fish_diseasehealth_management

27. Kruchten P. What Is the Rational Unified Process?. The Rational Edge. 2001. Available from: https://www.researchgate.net/publication/220018149_The_Rational_Unified_Process--An_Introduction
28. Anwar A. A Review of RUP (Rational Unified Process). International Journal of Software Engineering (IJSE). 2014;5(2):24.
29. Holden R, Boustani M, Azar J. Agile Innovation to transform healthcare: innovating in complex adaptive systems is an everyday process, not a light bulb event. BMJ Innovations. 2021;7(2):499-505. Available from: https://innovations.bmj.com/content/7/2/499
30. Kannan V, Basit M, Youngblood J, Bryson T, Toomay S, Fish J et al. Agile co-development for clinical adoption and adaptation of innovative technologies. 2017 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT). 2017;:56-59. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6197812/
31. Sindhwani R, Singh P, Prajapati D, Iqbal A, Phanden R, Malhotra V. Agile System in Health Care: Literature Review. Lecture Notes in Mechanical Engineering. 2019;:643-652. Available from: https://www.researchgate.net/publication/332602412_Agile_System_in_Health_Care_Literature_Review
32. Karlsson J. Agile Product Backlogs | Perforce Software. Perforce Software. 2022. Available from: https://www.perforce.com/resources/hns/agile-product-backlog-basics
33. Sedano T, Ralph P, Peraire C. The Product Backlog. 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE). 2019;41. Available from: https://www.researchgate.net/publication/330823863_The_Product_Backlog
34. Pinchas A. 9 Techniques for Prioritizing the Product Backlog | Parabol. Available from: https://www.parabol.co/blog/product-backlog-prioritization-techniques/
35. 7 Ways to Prioritize Your Product Backlog. Productplan.com. 2022. Available from: https://www.productplan.com/learn/prioritize-product-backlog/
36. Yarlagadda R, Hussaini Syed H. SOFTWARE ENGINEERING AUTOMATION IN IT. INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH

AND TECHNOLOGY. 2021;8(2):54-62. Available from: https://www.researchgate.net/publication/351853523_SOFTWARE_ENGINEERING_AUTOMATION_IN_IT

37. Bhagat V. Automation in Software Development- A New Stream of Revolution. PixelCrayons. 2021. Available from: https://www.pixelcrayons.com/blog/automation-in-software-development-a-new-stream-of-revolution/
38. Francis R. The 7 worst automation failures. CSO Online. 2022. Available from: https://www.csoonline.com/article/3188426/the-7-worst-automation-failures.html