NAME: THEOPHILUS G.N.M.MARTEY ATTUQAUYE

TITLE: SOFTWARE ENGINEERING

CODE: A2020CSD5.5M001

Assignment 1

NASA's Mars Climate Orbiter

On its mission to Mars in 1998 the Climate Orbiter spacecraft was ultimately lost in space. Although the failure bemused engineers for some time it was revealed that a sub-contractor on the engineering team failed to make a simple conversion from English units to metric. An embarrassing lapse that sent the $125 million craft fatally close to Mars' surface after attempting to stabilize its orbit too low. Flight controllers believe the spacecraft plowed into Mars' atmosphere where the associated stresses crippled its communications, leaving it hurtling on through space in an orbit around the sun.

The Mariner 1 Spacecraft

On a mission to fly-by Venus in 1962, this spacecraft barely made it out of Cape Canaveral when a software-coding error caused the rocket to veer dangerously off-course, threatening to crash back to earth. Alarmed, NASA engineers on the ground issued a self-destruct command. A review board later determined that the omission of a hyphen in coded computer instructions allowed the transmission of incorrect guidance signals to the spacecraft. The cost for the rocket was reportedly more than $18 million at the time.

The Morris Worm

A program developed by a Cornell University student for what he said was supposed to be a harmless experiment wound up spreading wildly and crashing thousands of computers in 1988 because of a coding error. It was the first widespread worm attack on the fledgling Internet. The graduate student, Robert Tappan Morris, was convicted of a criminal hacking offense and fined $10,000. Morris's lawyer claimed at the trial that his client's program helped improve computer security.

Costs for cleaning up the mess may have gone as high as $100 Million. Morris, who interestingly co-founded the startup incubator Y Combinatory, is now a professor at the Massachusetts Institute of Technology. A disk with the worm's source code is now housed at the University of Boston.

Assignment 2

SOFTWAER ENGINEERING MYTH

1. **Software Development Comes with a Hefty Price Tag**

Perhaps this is the most popular myth about software development. It is because of this myth companies do not harness the potential that custom software can provide which can improve their organization's efficiency. Instead, they opt for purchasing some "one size fits all" solution which, of course, doesn't fit their requirements and they have to find other means to work around it.

Also, consider the investment loss if the company outgrows the software and it just becomes unworkable. If you combine this with hidden costs such as upgrade fees, licensing and support costs, custom software doesn't seem so expensive.

2. The Waterfall Method Still Works

You would be amazed at the number of people who still believe that a system can be specified in detail, before you even build it. Not only is this almost impossible, but it is also inefficient to execute the development process in a sequence. While there also exists a myth that Agile lacks any planning whatsoever, the fact of the matter is that planning is just as necessary to the effectiveness of Agile as it is to Waterfall, but the difference is in the way the planning is done.

Waterfall promotes planning before building at the very beginning of the project that poses a lot of constraints in the flexibility and adaptation. Conversely, Agile allows for ongoing planning mechanism where changes and adjustments are made as the project goes along in an iterative manner.

Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

For Example**,** A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.
Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

Assignment 3

Software Development Life Cycle Models

A software life cycle model is a descriptive representation of the software development cycle. SDLC models might have a different approach but the basic phases and activity remain the same for all the models.

1. Design

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

2. Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

3. Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer's expectation. Testers refer SRS document to make sure that the software is as per the customer's standard.

4. Deployment

Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

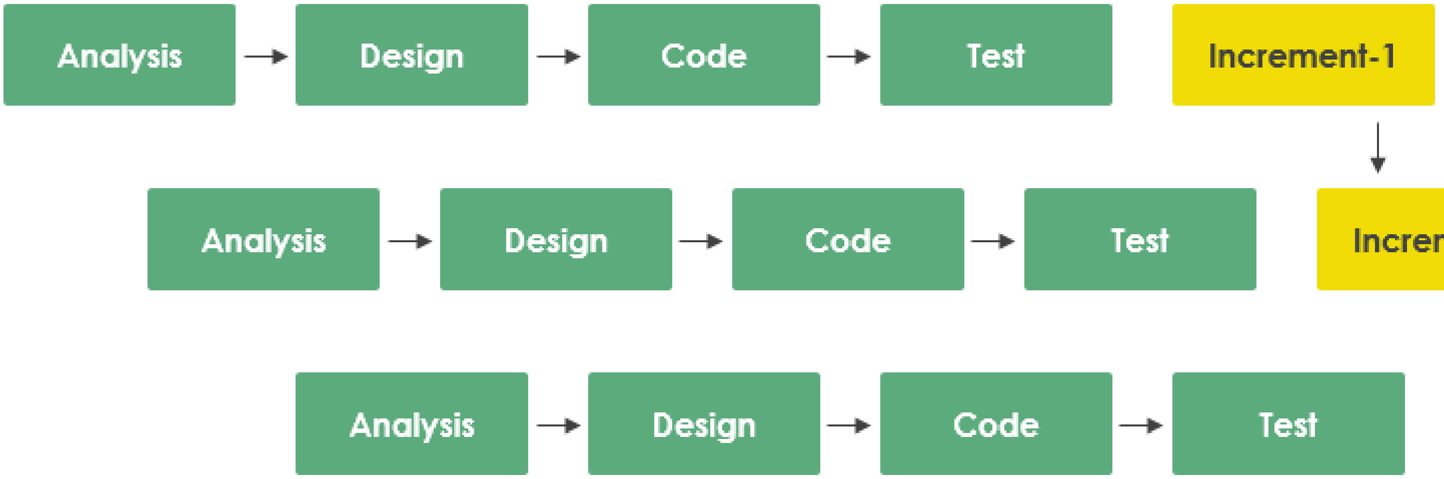
5. Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

ASSIGNMENT 5

Incremental model

The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all designed functionally has been implemented. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.



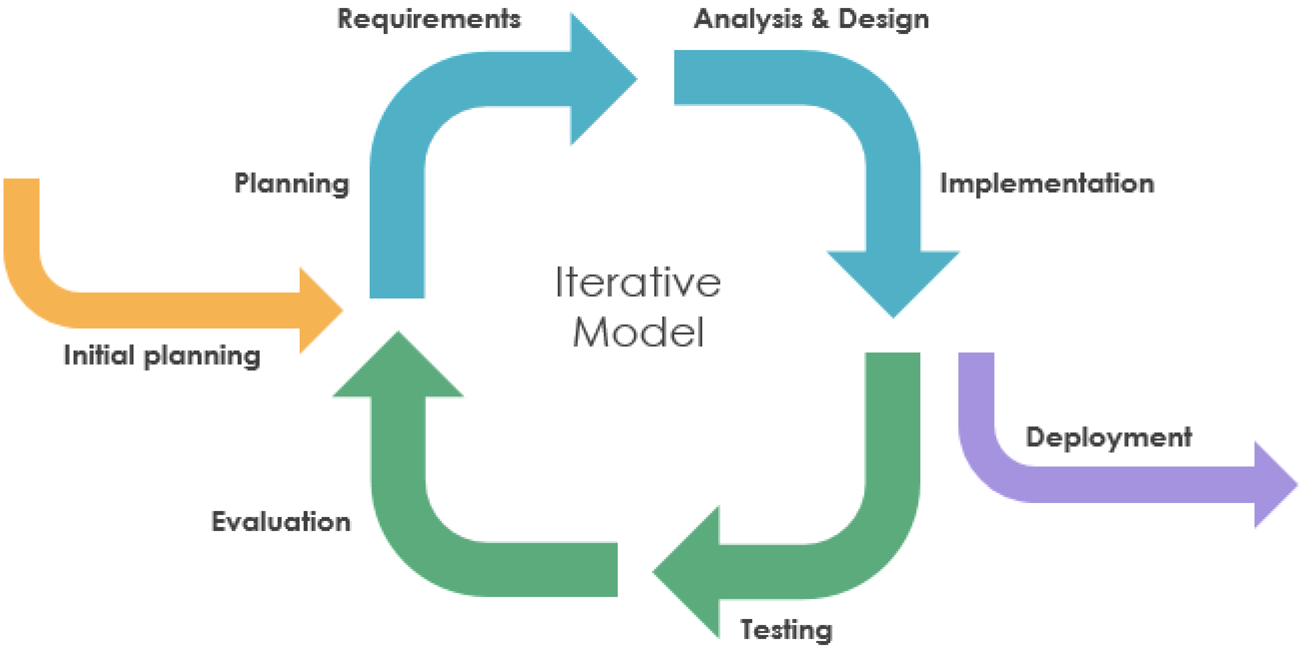
Incremental Model

1. Iterative Model

An iterative life cycle model does not attempt to start with a full specification of requirements by first focusing on an initial, simplified set user features, which then progressively gains more complexity and a broader set of features until the targeted system is complete. When adopting the iterative approach, the philosophy of incremental development will also often be used liberally and interchangeably.

In other words, the iterative approach begins by specifying and implementing just part of the software, which can then be reviewed and prioritized in order to identify further requirements. This iterative process is then repeated by delivering a new version of the software for each iteration. In a light-weight iterative project the code may represent the major source of documentation of the system;

2. However, in a critical iterative project a formal software specification may also be required.



Assignment 6

A **Functional Requirement** (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional Specification.

In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behavior of the system.

Assignment 7

Artifact of Software Engineering

During the process of software development, many tangible by-products are created. These are called artifacts. Artifacts play a vital role in the software and systems development processes. Here’s an in-depth discussion of what an artifact is and why it’s important in the software development process.

An **artifact** is one of many kinds of tangible by-products produced during the development of software. Some artifacts (e.g., use cases, class diagrams, and other Unified Modeling Language (UML) models, requirements and design documents) help describe the function, architecture, and design of software. Other artifacts are concerned with the process of development itself—such as project plans, business cases, and risk assessments.

The difference between specification and requirement

A Requirement is a statement of one thing a product must do or a quality it must have. A Requirement Specification is a collection of the set of all requirements that are to be imposed on the design and verification of the product. The specification also contains other related information necessary for the design, verification, and maintenance of the product.

The entire project depends on requirements. The first step to develop a software is to do a feasibility study. It focusses on technical aspects of the product. Next process is to gather requirements. It is possible by communicating with clients, end users and system users who will use the product at the end. Interview, surveys and questionnaires are main methods of collecting requirements. Finally, analyzing occurs after requirement gathering

Nowadays patients want to get professional medical care as well as quality service. The following features hospital patient management software among others can help to increase the level of customer service:

- Setting up the automatic processing of incoming calls from clients allowing the machine to address common questions without hospital staff involvement.
- Displaying the information about visiting hours and calendar slots available for doctors' appointments on the clinic site enabling patients to choose their convenient time and book an appointment online.
- Sending SMS messages or voice reminders to patients about the appointment time and share other important information about their treatment.
- Automating the process of filling out medical records and speeding up the patient registration process.
- Sending the results of medical tests and examinations to the patients via email.
- Monitoring the patients' satisfaction through easy-to-use feedback forms or NPS tracking.

The medical practice management system will help doctors to devote more time to patients, not documentation:

- All the necessary information about the patient is collected in one place and so any medical specialist within the clinic is able to access the full history of the disease and test results and, so, get more information to diagnose more accurately and prescribe the right treatment.
- Creating the database of document templates helps the doctors and nurses always provide a standardized medical protocol to the patient and quickly add information to the patient's card, which optimizes the duration of an appointment without compromising the quality.
- Doctors are in full control over their schedule and can arrange the workload efficiently.

Medical practice management software allows hospital authorities to take care of the profitability of the facility and get insights into streamlining internal processes of the clinic:

- Analyze doctors’ schedules and patients base and evaluate the distribution of the workload;
- Optimize hospital peak hours, reduce medical equipment downtime;
- Discover which medical directions generate revenue, and which should be closed or modified;
- Evaluate the visitors' satisfaction and find new ways to enhance it;
- Control the finances and maintain profitability.

Let's do a brief overview of the possibilities of automation in detail.

FEATURES OF THE HOSPITAL MANAGEMENT SYSTEM

Basically, medical practice management system consists of 3 layers:

1. Automation modules. Modules are used in daily operations of the clinic: doctors work with the patient's electronic health records and treatment plans; receptionists use the patients' registration and finances management module; the clinical laboratory interacts with the module storing all examinations and test results of the patients, etc.
2. Catalogs and protocols knowledge base. All the internal medical protocols and process standards, as well as patients' healthcare data - all the data that is required to enable the health care delivery and hospital administration.
3. Reports and dashboards. Administrative and medical reports and dashboards, that allow for analyzing the efficiency of the medical services provided and generate insights into improvement.

PATIENT REGISTRATION AND ELECTRONIC HEALTH RECORDS (EHR)

One of the integral parts of any clinic management part is basically a CRM that automates all the use cases relevant to patients interacting with different kinds of medical specialists within the clinic. The system should store as much information as possible about the patient - everything from personal data, insurance, and financial details up to the family health history that could suggest a better treatment approach to the doctors.

To ensure an individual approach, this data could provide the basis to create unique offers, certain services bundling and discounts to match the needs and expectations of every patient and their families.

It is also a great option for the healthcare facilities to share the information about a patient that has consulted with different institutions: complaints, examination results, lab tests, treatment plans and outcomes, history of the disease, the current status, diagnosis, available for doctors across the facilities allows for a more profound understanding of the health status of the patient. Therefore, integrations with third-party databases give very feasible benefits for almost any medical facility.

BILLING AND PAYMENT PROCESSING

All provided services should be chosen at once and also the visitors' payment amount should be automatically calculated in the program when making an appointment. It is supposed that the payment amount is already known when the patient applies for certain services at the reception.

The system should automatically track the debts or prepayments of the patients. The system feature of marking patients that have already paid the services or those who have to pay can be added. At the patient's request, it should be possible to print statements including a list of all services provided and the number of payments for the required period.

APPOINTMENT AND RECEPTION MANAGEMENT

The hospital scheduling software can divide the timeline into necessary periods for each doctor (for example, an average appointment duration can vary from 30 minutes to one hour depending on the doctor). This monitoring program can be easily customized for any type of specialist.

Hospital applications should allow patients to make appointments based on the availability of the needed rooms or medical equipment. If the patient cancels an appointment, the doctor should be notified and other patients could be notified about the new available slot.

The monitoring program should include reminders for the patients about regular checkups. It will help to control the patient's health.

Registration also allows seeing the number of patients visiting each doctor in the hospital.

LABORATORY AND TESTS MANAGEMENT

The clinic should be able to keep its own records as well as to form plans according to the international standards. The hospital management system should store the results of all laboratory research in electronic form.

Clinic mobile management app can keep records for each doctor and make payrolls, depending on various factors. It can be a number of appointments, the number of referrals to other specialists of the clinic, revenue, etc.

INVENTORY MANAGEMENT

The hospital inventory management module helps to keep track of the materials flow, automatically calculate the goods in the warehouse, and manage procurements.

The clinic management app enables the automatic write-off of the needed materials during the procedures.

FINANCE AND ACCOUNTING

Financial accounting automation helps to allocate funding easily, accurately, and regularly for all hospital departments. These processes can be monitored by any authorized users.

The accounting module of a hospital management software creates a unified system by automation of the full cycle of the medical facility operations from the reception to the laboratory.

Also, mechanisms of forming any combined financial statements by departments, employees, or services should be provided.

INSURANCE SERVICES INTEGRATION

The insurance module can record and store the patients' insurance details: policy number, the insurance company, information about policies. In this respect integration with third-party insurance applications could be important and useful allowing the patients to manage their payments and policies.

TREATMENT PLANS MANAGEMENT

Software for hospital management can allow doctors to select a diagnosis according to the ICD (International Classification of Diseases) or develop their own treatment plans based on several protocols. When maintaining the management system of treatment protocols, the hospital program will advise on how to examine and treat a patient effectively based on internal knowledge database of treatment outcomes.

Depending on the needs of the specific specialists, hospital management software can include the ability to compare the plans of medical examination and treatment, prescribed by a doctor with the plan in the treatment protocols against the outcomes for the patient.

REMOTE PATIENT MONITORING

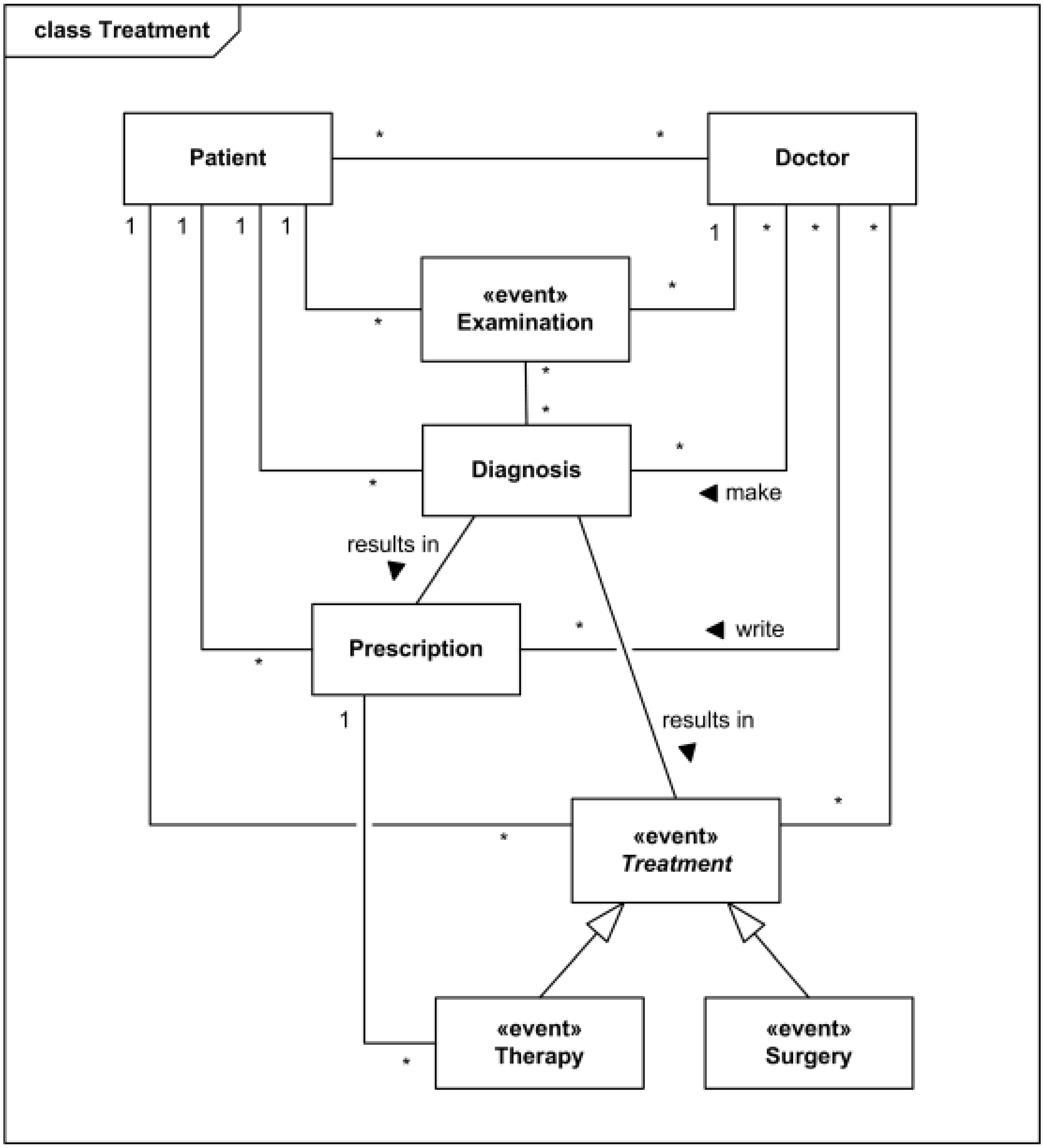
The remote patient monitoring feature in the clinic management software can inform the doctor about medication intake frequency and contraindications, gather basic data about patients' health conditions from the wearable devices or urge patients to manually enter the details about their wellbeing.

TELEMEDICINE

The features enabling long-distance doctor visits and support offer a significant added value for the patients. The telehealth functionality allows for a more individual approach to treatment, a more complex understanding of the patient's health conditions, and treatment outcome.

PHARMACY MANAGEMENT

In the hospital management system, it is necessary to monitor drug expiration dates in the stock. Also, the control module is needed, which helps to plan the delivery of medicines and products.



Facebook Post - 32 Hospital management system components.png
archer hospital management system components

REQUIREMENTS FOR HOSPITAL MANAGEMENT SOFTWARE

Let's look at the basic requirements to take into account when developing a proprietary hospital management solution:

1. Any employee of the clinic must be able to cope with the clinic management system.
2. The user interface must be robust, easy-to-use, convenient, and informative not allowing any misunderstandings as to the data being entered into the system.
3. There should be certain role-based access for each employee according to the authority and responsibilities. Hospital authority must be able to edit the access role and passwords of the employees. It will allow control of all confidential information.
4. Hospital software has to provide all the accurate, necessary, and relevant information at any time to the relevant specialist in the most convenient and secure format.
5. As we are talking about patients' personal data, the maximum level of security and data protection must be provided in the system, the software should be built in compliance with HIPAA or similar standards of healthcare data security.
6. The architecture of the medical management software should allow for scalability and extending the functionality to improve the clinic process automation and health care delivery.

BOTTOM LINE

Software for medical institutions must fully meet the patient's needs and improve employees' productivity at the same time. The high-quality system significantly reduces medical paperwork, improves the quality and informational content of medical documents. The software also helps to reduce the influence of human factors, and the process automation minimizes forgetfulness and inattention of the staff when registering patients and providing the consulting services.

With all these issues in mind, we can see that the introduction of new technologies into the hospital infrastructure is a critical issue for today. If you are looking for more information about how to develop a hospital management system or want to build a custom IT solution for your business, contact our team.