CS5030 – Software Engineering Principles – Exam

Matriculation Id – 220032472

December 07th, 2022

1. Software lifecycle and Processes:

   1.a. Three functional requirements of the new software developed to manage logistics of a busy hospital are:

Id	Functional requirement description	Priority	Source	Dependency
1.	Patients must be able to **register** with the new system, and must be able to view their **existing record** and **appointments**	High	The existing system must provide patient record and their past appointments	
2.	Staff must be able to register to the system and view their **records, work schedules,** and **access relevant functionality**	High	The existing system must provide staff related information and their work schedules	
3.	The new system must be able to manage the following requirement 1. **Patient record** 2. **staff records** 3. **patient appointment schedules** 4. **staff schedules** 5. **Prescriptions** 6. **Stock and inventory management** 7. **Hospital facility booking**	High	The existing system that handles these management, must provide data to new system	Some of the data fetched in the above two functional requirements (1 and 2) helps in constructing the management system for patients and staffs.

Three Non-Functional Requirements for the given system are:

1. **Availability** – The system must be available for access 24 *7 by the patients, staffs or other hospital members, to view any of the required information, such as patient records, staff work schedules, stocks, or prescription respectively. The availability of these information at any given time is crucial, as the system is built for a busy hospital, which operates on time bound emergency scenarios and cannot afford to lose time due to unavailability of information.

2. **Reliability** – The system must be reliable at any given time and must correctly deliver information as expected by the user. Some of the information provided by this system - stock management service for example, needs to be accurate and reliable, as it can lead to serious consequences that can endanger human life in case unreliable information.

3. **Security** - The system must be able to resist accidental or deliberate intrusions. The patient record, staff work schedule or prescription are personal and confidential information and cannot be made available to intruders. The system must have authentication and authorization in place before providing any such information to the requester. A case of security breach can lead to damaging reputation for the hospital.

1.b Software development process

As the system being built is **critical**, as it holds sensitive information and need to evaluate and validate end to end working without any misses in terms of security, safety, availability or resilience of the system, the **Modified Waterfall methodology** can be utilized to implement this software. All the requirements must be gathered in advance and they are less likely to change over period of time, as most of the system functionality are already present in multiple systems and they are tested and validated before. Hence with the prior availability of all the design and requirements elicited, the modified waterfall can suit the development of newer system. The system also

1.c. Layered or Tiered Architecture

As stated in the non-functional requirement, security is one of the concerns with respect to building the system for hospital hence **Layered Architecture** can be considered, where the most important critical asset (data holding records of patients, staffs or inventory) must be protected in the innermost layer and high-level security validation applied to these layers.

Each of the information (patient and staff records) can be encrypted and authorized before being served to the user.

The newer system is spread across varied user or personas (patient, staff and hospital management) and the application is to be built with multiple functionalities for each of these users, then such requirements can be separated and formed into layers of functionality and individual teams can work on each of these functionalities for each persona. This is possible in layered architecture as it supports separation of concerns, and each functionality can be built independently and integrated at later stage.

Since the data security is being focused in this case, each layer can be provided with authentication in order to improve dependability of the system. All data must be kept private with strict control access.

The layered architecture will consist of the following layers

1. Presentation layer – This is the topmost layer that is viewed by different users of the application (patients, staff or hospital management). This layer is used to view patient record information, appointments, staff schedules, inventory list, etc. as per the request of the respective users. This layer is responsible for taking commands form the user and passing it to the logic layer, where all the validation and data transformation takes place.

2. Business logic layer / data access layer – This layer performs all the authentication, authorization and data validation received from the user, and critical business problems are solved and presented to the above presentation layer. For example, in case of stock management, updating the data when a stock order is placed in the corresponding data storage, or to validate patient appointment with doctors based on their availability schedule, or to construct a hospital building maintenance plan for staffs, is all performed in this layer.

   As stated in the requirement, that all the functionalities are already present in terms of separate systems, we can reuse logic with added security feature in this layer (with some refactoring if needed) . This layer is also responsible for maintaining consistency and data structuring when integrated with the older system.

3. Data layer: This tier refers to the database that holds all the required information for the system. As there already exists a database with all the necessary information, it is easier to incorporate it into the newer system. The information is retrieved and updated by the logic layer mentioned above, which is then eventually transferred to the requested user via presentation layer.

1.d. Testing strategy:

The following test strategies will be employed to check for proper working of the implemented new hospital management system:

1. **Unit test** – All the individual functionality must be tested during development with the help of unit test.
   The test must be able to replicate the following scenarios
   1. Legal input – When an expected legal and correct input is provided by the user then the method or function must work as expected
   2. Illegal input – when an input is not of the expected form, the system must not crash and must provide appropriate response.

   Simple AAA testing strategy can be utilized for unit testing and automation

   Arrange – Setup the unit (method or function), with required data (input)

   Act – Call the object or method that needs to be tested

   Assert – Validate if the given input produces the expected output

2. **Component testing**– All the individual functionality (sub system) must be integrated after their development and must be tested for proper interaction. This test is done with the assumption that all individual functionalities are unit tested and the result is asserted as expected.
   The test must be able to provide the following result
   1. Component work as expected when integrated with other units as expected, mimicking the behaviour of the existing system.
   2. All the subsystems work as expected post integration.

3. **System testing**– Focuses on testing the interactions between all the components, developed by different teams or subsystem. It be tested for expected working after integration. This refers to testing of the whole system in place. The test must check the following:
   1. Component are integrated as per the specification
   2. Components are compatible with each other and are interacting as expected.

2. Software Quality

2.a Dimensions of dependability

The current system must ensure the following dependability when building new algorithm:

i. Error tolerance : This property can be considered as part of usability and reflects the extent to which the system will be designed, so that errors are avoided and tolerated with respect to calculating the fraud score. When user errors occur in

calculating the fraud score , the system should, as far as possible, detect these errors and either fix them automatically or request for newer set off transaction history.

ii. Reliability - System is unreliable if the data is hacked or corrupted from external attack. Data that is used to calculate the fraud score i.e., combination of data about card holder and the user's transaction history, must not have been tampered and only reliable data must be used to build the current system. Since the newer system is being built as the customers had already raised concerns regarding genuine transactions being flagged, the reliability of the new system must be higher importance.

iii. Maintainability – As stated in the requirements that the newer system will be using artificial intelligence techniques, which would provide better results with distinct data and large data set, the system must be able to allow for adaptation to new changes. The algorithm should be able to accept changes and evolve as the application usage grows.

2.b. Ethical concerns

1. Fairness of algorithms – The algorithm produced using artificial intelligence can be susceptible to bias. There have been incidents recorded in the past where some of the models generated for building AI algorithms were not neutral and provided results in an unexpected way. As these algorithms are written by humans, there were chances of bias, when building them. As a good software ethics, the system must be fair and must be built only based on the combination of user data and transaction history, without the consideration of the background, ethnicity, race and other personal factors of the user, or being biased.

2. Data privacy and security: All the data collected for the betterment of the AI algorithm must be transparent and indicated to the customer beforehand, and the information collected must be used only to track the efficiency and improvement of the newer system. Improper usage of this data may lead to privacy infringement

2.c Security threat

The following security threats are possible in the newer system

1. Interruption : The normal working of the system might be prevented by making some part of the system unavailable. There is a possibility for denial-of-service attack to be placed on the newer system. This can prevent in proper calculation of fraud score.
2. Interception: The Attackers can access to the newer system and leak the user information collected for betterment of the fraud score calculation algorithm.
3. Modification : As the system is relying on the past transaction history of the user along with the user data, any modification or tampering made to this data, can lead to failure and on the correctness of the result produced by newer system

4. Fabrication – Inserting false information into the system can be critical issue in case of this system, where false transaction can be added to the system and it might result in incorrect score.

2d. The Security concerns in the modified system:

Requirement : The security policy should set out when discussing the requirements of the modified system. This can include topics like “what is expected of customer to prevent unauthorized user or intrusion to their account”. One solution would be to use strong passwords for their accounts, and setup multi factor authentication when accessing and using their accounts. This can help in preventing transactions that are not done by the customer and can be flagged immediately to the system. This can also help in making the system understand an illegal transaction and prevent it from using for computing fraud score.

Software architecture: The architecture must consider the security and safety of the system. A distributed architecture can help in minimizing the effects of attacks on the system. This can also prevent single point of failure.
Performance - adding security check slows down a system, so its response time or throughput may be affected. This must be compensated by the architecture.

Design: When designing the new system, the developer, must log all the user actions to provide transparency, and allow for validation against the customer.
This can help in correctness of the data provided to the system.

3. Project Management and Collaborative development

3.a The generic project management activities are:

1. **Project planning** – This activity refers to planning, estimating and scheduling project development, and assigning people to tasks.
At the beginning of the project, at this phase, **cost estimates** and other needs for the client are drafted.

2. **Risk management**- The possible risks in building this suite of application, that may affect the development process must be analysed and monitored along with actions taken to rectify the risks whenever they occur.

Example of a risk related to the project that can occur is

1. When a team member might leave during development affecting the schedule of the project development or release.
2. If a rival competitors product is released before the said release date of our product. This can lead to risk in our development.

3.**People management** This activity assigns individuals to the development team and establishes methods of operation that enhance team performance. Each team will comprise of people with varied skills.
Since the teams are separated by geography, need to standardise on the work timing, as they might be from different time zones and might have to decide on the work handoffs between members of the team.

3.b Project management Challenges

a. Project scheduling – Estimation and task allocation for each team member might be a challenge. Need to identify and estimate the following:

a. Calendar time needed for each task – considering the time zone of each team member
b. Efforts required for a particular identified task
c. Who works on the task that has been identified in the project planning.
d. Apt resources needed to complete the task

b. Scope Creep – Since the scope of this application is large, the tracking and management of all the scopes can be difficult, and with additions of any new feature it can lead to dilution of the original requirement.

3.c. Category of risk

i. Technical Risk : This risk is identified with respect to the development of suit of programs. The functional requirements state to develop multiple applications performing individual functions and allow for seamless interaction between the system. There can be a risk when integrating all the systems together and allowing for interaction among them.
Example – The feedback and individual assessment data of student must be utilized to give the student journey through the program. In case these elements don't interact as expected, then the system is prone to misinformation.

ii. Project Management Risk – As the teams are present in different geographical local, each member come with different cultural and ethical backgrounds, with varied skills. There can arise a conflict on basis of culture and morale of the team and such interpersonal issues could impact results. If no work policies are set in place before the start of the project, it can lead to serious disruption of work.

iii. Organizational risk – The resources and staffs must be available to cover the time and effort needed to complete the entire suite of programs. If any team hasn't completed its development, then it might lead to problems when assemblies the entire system.
Example – if individual teams are assigned to build system to deliver live classes, and online support, and if the system is built at different time, then

the complete testing of this system as whole, where the student can interact and get support during live classes cannot be done.

3.d. Strategies to ensure that team work well together

Motivation : As a manager one must motivate the team members working on the project. Encouragement leads to work efficiency and better performance. Lack of motivation results in disinterest in the job, decreased productivity, increased error rates, and failure to reach organisational objectives.

Independent decision-making - Trusting the team in making their own decision. People frequently develop resentment toward their work if they lack the autonomy and authority to make decisions on it, which hinders the efforts of the entire team. As a manger, one must trust in the collective decision made by the team.

Demarcating Roles and Responsibilities: As the teams are geographically separated, there are confusions that can lead to conflicts, if one is uncertain about the role and responsibility of each member in the team. To prevent this, each member must be a setting specific objective which will keep everyone focused on their allocated duties. This will lead to unified progress.

Communicating openly and freely: Manager should be clear and important to inform the message in terms of any failures. The message should be in a constructive way to make the announcement, but not in a hurtful manner. Message must be conveyed without hurting the feelings of the members, or in a subtle manner with all the reasons for the failures covered. This leads to openness and trust between the manager and the team member, and clear understanding of the progress of the project.

Avoid Micromanagement – The team must be allowed to work freely without someone always looking over their shoulders. Team must feel independent and establish their own working style and deadlines (but must confer within the project deadline) when it comes to working as a team.