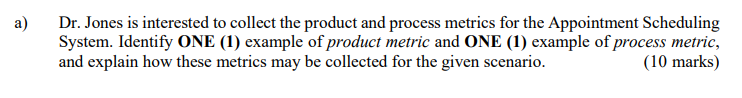
**Question 1**



| Product metrics  Answer 1 :  Dynamic metrics :   * ***Efficiency*** refers to the amount of effort and time that the users need in order to achieve their goal.   Measurement :   * ***Time spent performing and searching for a patient's record***. * The lower the time taken to obtain the patient’s information will be better as this indicates that the appointment scheduling system is efficient which is able to improve the workflow between clinics’ staff and complete a task in a short period of time.   Answer 2 :  Static metrics :   * **Line of code** which refers to the number of lines of the source code in a program. It is used to access the size and complexity of the software. It is relatively simple to be measured and understood.   Measurement :   * It can be measured by counting the number of lines in the program source code. The more lines indicate the system is more complex thus can be more error prone (i.e more bugs). Hence, the bugs need to be rectified before releasing the appointment scheduling system for operational use. |
| --- |

| Process metrics   * Private metrics which are private to the individual software engineer and serve the result as an indicator for individuals only.   Measurement :   * ***Defect rate (by module)*** - Total number of errors/defects performed in the appointment scheduling system module which is developed by a software engineer during the code inspection activity / testing or debugging. * The lower defect rate will be better as this indicates that the developer team has carried out sufficient testing before releasing the system for operational use. |
| --- |

b.

The suitable process model is the prototyping model. A prototype can be developed to give end users a concrete impression of the system capabilities.

Justification :

* The requirements given by the clinics’ staff is unclear. Hence, a prototype can be quickly designed and presented to the customer to elicit the requirements.
* The clinics’ staff is unfamiliar with the computer system in which their IT literacy is low. Through prototypes, it can also provide preliminary training to them from time-to-time.

WBS for prototyping :

| 1.0 Project Planning  1.1 Define Project Vision and Goal  1.2 Define Project Objective  1.3 Define Expected Project Benefit  1.4 Define Project Constraint  1.5 Define Stakeholder  1.6 Project Risk Analysis  1.7 Generate Procurement Report  2.0 System Requirement Analysis  2.1 Evaluation and Analysis of Existing System  2.1.1 Evaluate and Analyze Existing System Performance  2.1.2 Evaluate and Analyze Existing System workflow  2.1.3 Evaluate and Analyze Existing Database  2.2 Fact Gathering  2.2.1 Interview the manager and staff of the grocery store  2.2.2 Observe the current business process  2.2.3 Distribute questionnaire to customer  2.3 Define System Requirements  2.3.1 Drawing Use Case Diagram  2.3.2 Collect User Requirement  2.3.3 Document User Requirement into System Requirement  3.0 System Design  3.1 Define System Module  3.2 Drawing UML diagram  3.3 Design Database / Drawing ERD diagram  3.4 Construct Database  3.5 Drawing Sequence Diagram  4.0 System Development  4.1 Prototyping 1  4.1.1 Prototyping Discussion  4.1.2 Develop Prototype  4.1.2.1 Account Module  4.1.2.2 Product Module  4.1.2.3 Shopping Cart Module  4.1.2.4 Order Module  4.1.2.5 Payment Module  4.1.2.6 Promotion Module  4.1.2.7 Product Management Module  4.1.2.8 Order Management Module  4.1.2.9 Sales Report Module  4.1.2.10 Inventory Module  4.1.3 Presentation of Prototype  4.1.3.1 Collect Customer Feedback  4.1.3.2 Refine System Requirements  4.2 Prototyping 2  4.2.1 Prototyping Discussion  4.2.2 Modify System Design  4.2.3 Develop Operational Prototype  4.2.3.1 Add On module  4.2.4 Presentation of Prototype  4.2.4.1 Collect Customer Feedback  4.2.4.2 Refine New System Requirements  4.3 Final Construction  4.3.1 Prototyping Discussion  4.3.2 Modify System Design  4.3.3 Improve Operational Prototype  4.4 Testing  4.4.1 Develop Test Plan  4.4.2 Unit Testing  4.4.3 System Testing  4.4.4 Load Testing  4.4.5 Stress Testing  5.0 System Implementation  5.1 Subscribe Cloud Computing Service  5.2 Execute Software in Cloud Server  5.3 Provide Staff Training on Use of System  6.0 Closure  6.1 Final Project Report  6.2 Final Project Presentation  6.3 Project Completed |
| --- |

**Question 2**

Actual Cost = 136,000

Planned Value = 142,000

Earned Value = 200,000 \* 0.7 = 140,000

Schedule Variance = EV - PV

= 140,000 - 142,000

= (2,000)

Cost Variance = EV - AC

= 140,000 - 136,000

= 4,000

Cost Efficiency : 140,000 / 136,000 = 1.029

Schedule Efficiency : 140,000 / 142,000 = 0.986

Comment : Cost efficiency is performing better as it is higher (1.029) compared to schedule efficiency (0.986).

(ii) 2 corrective actions

* **Streamlining the process**
  + Remove unnecessary activities so that it would not complicate the development process to reduce the time for a staff to complete the task.
  + Standardized procedures for every single task such as ensuring all the requirements specified by the user have been applied, the unit testing has been carried out and code has been pushed to the respective branch during the software development.
* **Reassigning task**
  + Find the suitable candidates with appropriate skills and switch the tasks among them as each of the team members has different strengths. For example, some are good at Python while some are analytical.
  + This can increase their productivity as the task assigned is matched with their skills which can lead to better quality work.

Software Quality Attributes 1 :

**Functional Suitability :**

* This is important as it refers to the ability of the Appointment Scheduling System to perform its intended function such as summary and detail reports generating, searching for a particular patient record.

**Measurement** :

* **Functional Correctness**
  + The degree to which the system provides correct results with the required degree of precision.
  + It could be measured by calculating the percentage of error rate by counting the number of errors occurring when the task is undergoing the testing process divided by the total task undertaken then multiplied by 100%
  + The acceptable percentage of error rate for each task should be smaller than 5%.
* **Functional Completeness** 
  + It is important for ensuring the set of functions covers all the specified tasks and user objectives.
  + 0 and 1 will be used to indicate the completeness such that 0 means failed and 1 means completed.
  + For example, if the module is able to achieve all the tasks given, then the module is said to be completed.
* **Functional Appropriateness**
  + The function facilitates the accomplishment of specified tasks and objectives, for example by presenting the user with only the necessary steps to complete the tasks.
  + It could be measured by comparing the number of actions and amount of times that the user took in performing the task for each individual function/method with the previous and improved version of the system.
  + To illustrate, if the total actions and amount of time taken to complete a task for the improved system version are lesser than the previous version, it means the functional appropriateness of that improved system is achieved.

Software Quality Attributes 2 :

**Performance efficiency** is one of the product metrics that shall be taken into account for the system. Performance efficiency assesses how well the Appointment Scheduling System in performing the user’s intended function. Besides, performance efficiency also refers to the Appointment Scheduling System use of resources which will affect the scalability and responsiveness of the system (Maayan, 2017). The type of resources used may include the CPU, memory, disk and configuration of the system.

Performance efficiency of the Appointment Scheduling System can be measured through several aspects which include the response time of the system when performing certain functions, number of concurrent users the system can handle as well as the CPU and memory usage under peak conditions. In order to measure the response time of the system, the system shall be tested under low, moderate and heavy load conditions. For instance, under low load conditions, the system shall provide a 1 second response time while under heavy load conditions, the system’s response time shall not exceed 3 seconds. Furthermore, the number of concurrent users the system can handle will be tested through increasing the number of users on the system incrementally until the system is no longer able to process the

user requests. As for CPU and memory usage, they shall be utilized as much as possible under peak conditions. For instance, a CPU that remains idle under peak conditions would indicate that the performance efficiency of the system is poorly optimized.

Software Quality Attributes 3 :

Security

* This is important as it ensures the appointment scheduling system is free of security breach (i.e accidental or deliberate intrusion). It helps to protect the confidentiality of the patients’ record, the exposure of the records will result in a loss of trust towards the General Hospital Diabetes Centre and subsequently cause customer churn as the system failed to provide adequate protection to their privacy data.

**Question 3**

1. Based on my analysis, the most relevant CMMI process model will be the continuous model. This is because the continuous model does not classify the General Hospital Diabetes Centre according to its discrete level, instead, it considers the individual groups of practices and assess the good use of good practices within each group processes. The centre can then come up with a process capability profile which displays its CMMI assessment capability level. From the process capability model, the centre can pick and choose the process for improvement based on their needs and requirements which are more flexible while still allowing the General Hospital Diabetes Centre to work within the CMMI framework.
2. Requirement are not clear

| **Risk name** | **category** | **impact** | **RMMM** |
| --- | --- | --- | --- |
| The project team members are not equipped  with appropriate skills to complete the project. | Staff size and experience | 4, the project team has not much experience with the Python language, thus it will affect their productivity badly and cause projects to be delayed. | **Risk Mitigation**:  - Register a Python course for the project team to learn about new skills, to ensure that all the team members have sufficient knowledge to complete the project.  **Risk Monitoring**:  - Monitor the attitude of the project team and motivate the project team to ensure that they are keen to learn new skills.  **Risk Management**:  - Use the contingency allowances to hire third party expertise that has experience in using new skills in order to lead the project team to achieve their goals. |
| Requirements are not clear | Customer characteristic, communication between development and customer | 3, Unclear requirements will cause the software to fail to meet the user requirements. Furthermore, the quality of the software will be compromised and would not meet the customer's expectations. | **Risk Mitigation:**  • A tool-drawn prototype using Visual Basic is presented to the users to identify the interaction flow of the system and let the project team elicit the requirements.  **Risk Monitoring:**  • Observe and record users' responses during the prototyping demo session.  • Examine and ensure there is no requirements conflict emerging between users and the project team before proceeding to the actual system development.  **Risk Management:**  • Tool-drawn prototyping will be used to obtain feedback from the stakeholders. The feedback gathered will be incorporated into developing and improving the system.  • If any issue arises because the users' requirements are being incorrectly identified, the contingency allowances and reserves will be used to hire third-party experts to solve the problem to ensure the project can be completed within the schedule without delay. |

**Question 4**

**Rate of Fault Occurrence (ROCOF)**

* In order to measure the ROCOF of the system, we will count and record the number of failures occurring when performing a certain function during a prescribed time period. For instance, 1000 times of payment check out operations will be carried out and the number of abnormal events will be recorded. The formula used to calculate ROCOF will be shown below:



* The smaller the value of ROCOF, the greater the reliability of the system. For instance, the value of ROCOF which is less than 0.01 indicates that 10 failures occur in each 1000 operational time units.

**Mean time to failure (MTTF)**

* MTTF refers to the average duration for a non-repairable system to operate before failure occurs. The predicted elapsed time between inherent failures of the system during normal bill payment operation. It could be measured by applying this formula:
  + MTTF = Total hours of operation / Total number of units
* The result of MTTF should be longer than the transaction length. It can be used to predict when the component is likely to fail and fix it before the failure occurs thus can reduce repair cost and minimize system downtime.

1. Yes, I agree with this statement. This is because the error can actually be corrected using built-in error detection and recovery before it causes a system failure. To illustrate, the system can detect the error and correct it to prevent system failure. Moreover, the error can be protected against by the built-in protection facilities such as protecting the system resource from system error which will consequently lead to system failure. Therefore, system errors do not necessarily cause system failure.
2. **Competition** - A new system will be necessary if its respective competitor has introduced the new features. For example, when the IOS product comes out with a new feature, the Microsoft product will also introduce similar new features to compete with its competitor. Otherwise, the company’s market share will be lost if they didn’t provide the features to the existing customer as the customers are not satisfied with the current services.

**Technical quality of the system** - If the system currently has many bugs which haven't been remediated, it is necessary to issue a fault repair release. Otherwise, it will affect the way in which the customers use the system and cause them to be frustrated and unsatisfied with the system. If the system only has minor system failure, it can be repaired by issuing patches, normally distributed over the internet and applied to the current release of the system.