1. **(40 marks)**

(a) The quality of the software development process is the most important factor in the success of a software project. Discuss in a balanced way this statement, giving two important reasons in favour of the statement, and two important reasons opposing the statement. (5 marks)

Answer

- Providing quality software will improve consumer / client confidence in your business which makes it vitally important. Quality software can help ensure that the project team work on a common goal and not just churn out sub-standard products.
- While quality is important in the SDP, it is not wholly responsible for its success or failure, time and budget are primarily responsible for this. Microsoft has demonstrated in the past that quality software does not necessarily mean that its products are successful, they have realised poor software previously and it has not resulted in it being a failure (they have released several patches to fix bugs). The elimination of all bugs in software is not possible.

(b) An application is to be developed that will run on a tablet or smartphone, and allow a family doctor (General Practitioner) to access patient records, including, for example, medication details and images such as x-rays. You are the software project manager for this project. Answer all the following questions. Relate your answers explicitly to this project.

i. Give two reasons in favour of, and two reasons against, the use of an Agile process for this project. (4 marks)

Answer

For

- It allows the development team to produce software that the client will actually use by consulting GP's on the development and gaining their feedback.
- It will ensure that it complies with Data Protection regulations because it will be holding medical records where patient confidentiality is essential.

Against

- It can take longer to complete the project because the GP's may want change after change.
- Documentation will not be of a high standard and trying to prove the security and confidentiality of the software may be difficult.

ii. Four general stages of a project are: Requirements and Analysis; Architecture and Design; Coding; Software verification and validation. State, with reason(s), and in decreasing order which two stages are most important for the success of this project. (4 marks)

Answer

- Software Verification and Validation should be the most concentrated on stage, due to the nature of the software. It will have confidentiality issues relating to patient information so we must ensure that this is maintained at all times. One security breach that gains media attention would tarnish the name of the business beyond repair and lead to countless legal issues.
- Requirements and Analysis is a close second as this will help ensure compliance with verification and validation. The team will need to ensure that the software will comply with all legal requirements regarding Data Protection and having well documented code and careful analysis of the system will help defend against possible legal and moral issues in the future.

iii. State, with reasons, what are the two most important non-functional requirements for this system. (3 marks)

Answer

- Security, as previously mentioned it is essentially important that Data Protection and Patient Confidentiality are maintained.
- Quality is equally important to ensure that then patient records are accurate, it could be detrimental to the health of a patient should there be incorrect queries mixing up medication details for a patient. Therefore testing would need to ensure as close to 100% bug free as possible.

iv. State, with reasons, the single most important method of validation that should be done on this project. (2 marks)

Answer

-

v. State, with reasons, the single most important method of verification that should be done on this project. (2 marks)

Answer

-

(c) Explain how the Floyd-Hoare triple {P} C {Q} emphasizes the distinction between *what* a program is meant to achieve versus *how* the program achieves this. (2 marks)

Answer

•

(d) The following describes the syntax of a simple programming language:

```
C ::=  V := E
     | C1 ; C2
     | IF B THEN C1 ELSE C2
     | BEGIN V1; ... Vn; C END
     | WHILE B DO C
E ::=  N | V | E1 + E2 | E1 - E2 | E1 x E2 | ...
B ::=  T | F | E1=E2 | E1 <= E2 | ...
```

Describe how one can use Floyd-Hoare logic to formally prove that a *complete* program in this language satisfies a specification of requirements. (8 marks)

Answer

•

(e) Give an overview of Bertrand Meyer's Design by Contract (DbyC) paradigm, and discuss how DbyC relates to *defensive programming*. (5 marks)

Answer

•

(f) In a transport control system, a simple class represents a level-crossing barrier. This barrier class has attributes representing the state of the barrier: stopped, goingup, goingdown, isup and isdown. Write down a class invariant in the predicate calculus for this class. Add two operations (methods) to the class and specify them using a contract expressed in the predicate calculus. (5 marks)

Answer

•

2. **(40 marks)**

(a) A software system is under development to support online ticketing for cultural events. The system will support various functions including event browsing, ticket booking and purchasing. A UML-based software process following Larman's Agile UP (Unified Process) will be used in the development of this system. The following questions deal with activities of the Larman Agile UP.

i. Give the full use case text for one important use case, and construct a simple domain diagram (conceptual model) for this application. (5 marks)

Answer

•

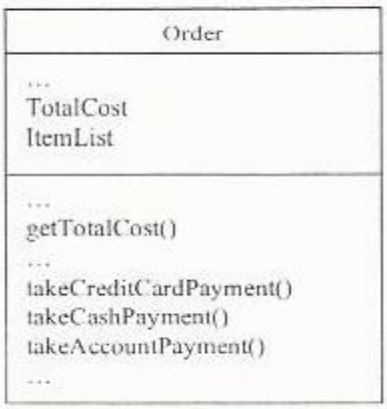
ii. Draw an appropriate **system** sequence diagram for this application, and describe what role **system** sequence diagrams play in the Agile UP process. (5 marks)

Answer
•

iii. Draw an implementation interaction (sequence or collaboration) diagram for this system. Describe how the interaction diagram relates to the other stages of the Agile UP process. (5 marks)

Answer
•

(b) A shopping software system includes an Order class representing a customer's order with attributes for the items in the order and the total cost of items being purchased. Methods of the class include various get methods, and methods for the different ways of taking payment for the order.



i. Evaluate the above design, stating two limitations it has. (5 marks)

Answer
•

ii. Illustrate with a diagram how the above design might be improved with a software design pattern (other than Factory). Name the pattern and state two benefits. (6 marks)

Answer
•

iii. How might the Factory software design pattern be used in the improved design? (2 marks)

Answer
•

iv. State two benefits of the use of Factory in the design. (2 marks)

Answer
•

(c) State two advantages of using statecharts rather than code as the basis for software development. (4 marks)

Answer
•

(d) Give a simple example of a statechart (e.g. for a software controller for a simple DVD player) to illustrate features of a statechart that are not found in conventional state machines (e.g. guards). Include in the statechart at least four features not found in conventional state machines, and indicate clearly these particular features. (6 marks)

Answer
•

3. **(40 marks)**

(a) Triple modular redundancy is often used to achieve fault tolerance in critical systems, such as aerospace control applications.

i. Explain this technique? (2 marks)

Answer
•

ii. State what types of fault it can deal with and state what types of fault it does not deal with. (2 marks)

Answer
•

iii. Describe another technique for fault tolerance, and state what type of faults it deals with. Discuss any limitations of this technique. (5 marks)

Answer
•

(b) What is meant by Six Sigma as used in statistical models of quality? What rate of defects does it correspond to? (4 marks)

Answer
•

(c) How relevant are Six Sigma statistical measures to software quality? Give detailed reasons for your answer. (4 marks)

Answer
•

(d) A program takes two integers, representing a year and month, and computes an integer result, giving the number of days in that month.

i. Without the source of the program, what method would you use to devise tests for the program? (2 marks)

Answer
•

ii. Give examples of the tests. (2 marks)

Answer
•

iii. Given the source code for the program, what other tests would you add? (2 marks)

Answer
•

(e) Explain, giving three reasons, why the testing of non-functional properties of software systems is difficult. (6 marks)

Answer
•

(f) How does statistical testing differ from defect testing? (2 marks)

Answer

•

(g) A software system has three modes: search; store; delete. The operational profile indicates that the probability the system is in the various modes is as follows:

search	store	delete
0.5	0.4	0.1

Historical data indicates that the probability of selecting a failure-causing input in each mode is as follows:

search	store	delete
0.0001	0.0001	0.005

i. For each mode, determine the number of random tests that must be run to have a 50% chance of detecting a fault. (6 marks)

Answer
•

ii. What information does this analysis give to the software developer? (3 marks)

Answer
•

4. **(40 marks)**

(a) A public transport reservation system is being designed that will include classes representing customers, journeys, trains, buses, schedules, bookings, tickets, payments, and include methods supporting various functions including journey planning, ticket booking, ticket purchase. The system can be accessed on a desktop, smartphone or street kiosk.

i. Outline a Model-View-Controller (MVC) architecture for this system. (5 marks)

Answer
•

ii. State two benefits of MVC for this system. (3 marks)

Answer
- It has a much more compact design allowing for less code to go through. It is not tied to any language i.e. it is independent of the code.
- It allows a broader view of the hierarchy, meaning it is more general.

iii. Explain Larman's basic software design principles of Indirection and Protected Variation and, using the above system, illustrate the principles with two design suggestions for this system. (6 marks)

Answer

- GRASP - (General Responsibility Assignment Software Patterns)
- Indirection
  The Indirection pattern supports low coupling (and reuse potential) between two elements by assigning the responsibility of mediation between them to an intermediate object. An example of this is the introduction of a controller component for mediation between data (model) and its representation (view) in the Model-view-controller pattern.
- Protected Variation
  The Protected Variations pattern protects elements from the variations on other elements (objects, systems, subsystems) by wrapping the focus of instability with an interface and using polymorphism to create various implementations of this interface.

Two Suggestions?

(b) Describe (with sketches if appropriate) how data and functionality are located in the repository, distributed objects, and pipe-and-filter architectures. (6 marks)

Answer

-

(c) An international financial institution maintains offices worldwide supporting local customers. Its information system must keep large amounts of data, including client (customer) data, stock market company profiles, and financial product information. In addition, the system must provide functionality for accessing and managing this data. Please ensure that your answers to the following questions are relevant to this particular project.

i. Describe (with a diagram if appropriate) a suitable (non cloud-based) distributed architecture for this application, and justify your choice, providing three reasons why it is suitable for this application. (5 marks)

Answer

-

ii. Describe two limitations of the architecture for this application. (2 marks)

Answer

-

iii. State the two most important non-functional requirements for this system, and discuss how well the chosen architecture satisfies each of these non-functional requirements. (4 marks)

Answer
-

iv. Give two examples of trade-offs among non-functional properties that can arise when implementing a distributed system. (4 marks)

Answer
-

v. Name an alternative (non cloud-based) distributed architecture model that might be suitable for this system, and describe two advantages and two disadvantages with respect to the previous architecture. (5 marks)

Answer
-

5. **(40 marks)**

(a) Why is asynchronous coupling important in cloud-based applications? (3 marks)

Answer
-

(b) An application has data stored in Blob storage on Google App Engine and needs to access a data processing application on Microsoft Azure to process the data. Give an overview (with sketch if appropriate) of a solution that uses asynchronous coupling. (6 marks)

Answer
-

(c) RESTful APIs are often used for web services and cloud-based applications. Give an overview of REST (REpresentational State Transfer), and state two reasons for adopting a RESTful API. (6 marks)

Answer
-

(d) Describe the two main options for scaling a (non cloud-based) architecture when faced with a large increase in number of users or transactions. (2 marks)

Answer
-

(e) Storing and accessing data poses challenges for large distributed global companies such as Facebook and ebay. Discuss in detail three of these challenges, and outline some corresponding solutions that might be adopted for these challenges. (9 marks)

Answer
-

(f) Certain properties of a distributed system are captured by Brewer's CAP conjecture (later proved as a theorem). State Brewer's CAP theorem, explain its significance and comment on how the CAP theorem influences companies such as Amazon in the design of their information systems. (4 marks)

Answer
-

(g) It is estimated that almost 90% of data storage bought in 2014 will be for unstructured data. Explain what is meant by unstructured data, and give one real-world example of unstructured data. (3 marks)

Answer
-

(h) A large file contains data on music downloads over an extended period, each line representing the number of downloads for a city on a particular day, e.g.
Cork 588
Given that this file has been split into ten equal sized files, describe (with a sketch, if appropriate) an efficient solution for finding the maximum daily downloads for each city using the MapReduce paradigm. (7 marks)

Answer

-