Course Title: Software Engineering.
Content: Examination Marking Guide.

Instructions

1. Answer question **ONE (Compulsory)** and any other **TWO** questions.
2. Write all your answers in the answer booklet provided.
3. Time allowed: **Two hours**.

Question #1 [30 Marks]

Consider following system development case:

JamboJet is in the process of acquiring an airline reservation System is a computerized system used to store and retrieve information and conduct transactions related to air travel. This application aims to have two parts. First is user part and the administrator part. User part is used as a front end and administrator is the back end. Administrator is used by airline authority. It will allow the customers to access database and allow new customers to sign up for online access.

Airline passenger should be able to search for flights that are available between the two travel cities, namely the "Departure city" and "Arrival city" for a particular departure and arrival dates. The system displays all the flight's details such as flight no, name, price and duration of journey etc. After search the system display list of available flights and allows customer to choose a particular flight. Then the system checks for the availability of seats on the flight. If the seats are available then the system allows the passenger to book a seat. Otherwise it asks the user to choose another flight.

To book a flight the system asks the customer to enter his details such as name, address, city, state, and credit card number and contact number. Then it checks the validity of card and book the flight and update the airline database and user database. The system also allows the customer to cancel his/her reservation, if any problem occurs. The system should then allow the administrator to generate reports of flight schedules, booking and cancelations.

a) Formulate a requirements engineering documents composed of clearly outlined:

- **User requirements**
- **System requirements**
- **Nonfunctional requirements categorized as product, organizational and external.**

[15 Marks]

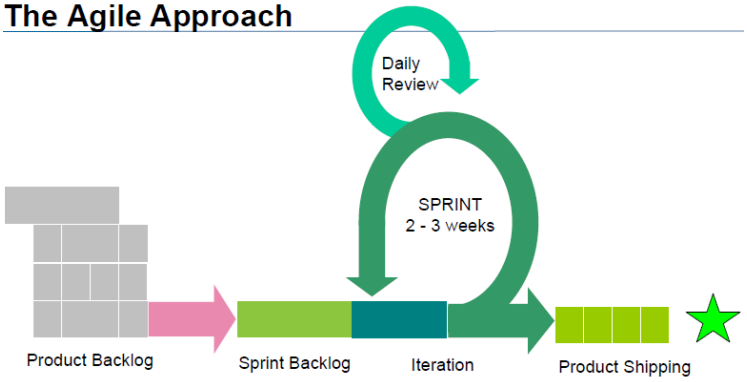
*open question. Marks to be awarded on how clear and correct the requirements are stated

b) Draw a use case diagram for the entire system. **[7 Marks]**

*the students should identify the main users (customer and administrator) with the associated use cases as depicted by the various functionalities outlines in the case. Standard symbols for drawing use case diagrams must be used.

c) Discuss how the scrum approach for agile project management could be used during the development of the system. Use diagram with clear explanations. [8 Marks]

*the scrum is an agile project management process model. The student should elaborate on the model with much emphasis on the sprint cycle. The explanation should indicate the fact that the entire problem is split into manageable features that get to the sprint cycles for 2-4 weeks with daily review meeting. The following diagram could be used to add clarity:



The student should state the fact that the sprint cycle has a sprint manager, developers, customer and stake holders. It involves four main phase: Assess, Select, Develop and Review.

Question #2 [20 Marks]

a) Discuss why the hybrid methodology is best suited for the development for the system. [6 Marks]

*either methodology is applicable. However the agile approach could be the best in this case. The student should clearly support his/her choice over the other one.

b) If you were contracted to develop the system, with the aid of diagrams supported by explanations, outline any two hybrid process models that you will opt to use. [14 Marks]

*any hybrid process model could be discussed. These includes:

- Incremental delivery
- Incremental development
- Scrum waterfall model
- Spiral hybrid (spiral+prototyping) etc

Question #3 [20 Marks]

a) What is software prototyping? [2 Marks]

Software prototyping is defined as a rapid software development for validating the requirements.

b) State and explain the two prototyping approaches in software process? [8 Marks]

The prototyping approaches in software process are:

- **Evolutionary prototyping**: In this approach of system development, the initial prototype is prepared and it is then refined through number of stages to final stage.
- **Throw-away prototyping**: Using this approach a rough practical implementation of the system is produced. The requirement problems can be identified from this implementation. It is then discarded. System is then developed using some different engineering paradigm.

c) Discuss the view that software prototyping with incremental delivery can be useful during the requirements analysis process. [10 Marks]

A good answer should:

- Present an overview of the commercial environment within which most software is produced, namely one of: strong competition; available and easily accessible productivity tools; and very knowledgeable and discerning consumers of software.
- Argue that, given the changeable nature of customer requirements, prototyping and incremental developments are key approaches by which software developers can reduce the risk of producing the wrong product on time, and the right product in an untimely manner.
- Recognise that the shortage of professional developers, and the lack of regulation to the profession of the self-taught developer, can give rise to the problem of poor quality and inefficiency in the use of resources resulting from excessive system fragmentation, poor performance, and concealed defects.
- Acknowledge that prototyping will still play an important role in eliciting requirements today and in the future, especially as user needs are often hybrid, and more complex.
- Further, the increasing adoption of software openness and reuse may limit the excesses of fragmentation and increase the reliability and longevity of systems.

Question #4 [20 Marks]

a) State and explain any three reasons why system modeling is important in software engineering. [6 Marks]

- As a means of facilitating discussion about an existing or proposed system

  Incomplete and incorrect models are OK as their role is to support discussion.

- As a way of documenting an existing system

  Models should be an accurate representation of the system but need not be complete.

- As a detailed system description that can be used to generate a system implementation

  Models have to be both correct and complete.

b) Define Reverse Engineering. State and explain three main objectives of reverse engineering? [7 Marks]

The reverse engineering is the process of generating representations that are implementation independent, starting from code. It is opposite of normal forward engineering process. The main objectives of the reverse Engineering process are:

i) It helps the companies to understand the complexities of the system

ii) Helps the analyst to generate useful lost information about legacy systems

iii) Can be used to identify reusable components for analysis and future use

iv) Helps in generating graphical representation of the system from different perspectives e.g. ER diagram, DFD, class diagram etc.

v) Can be used as a part of Reengineering process.

vi) Over a period of time modifications made to the software also result into unexpected problems. The anomalies can be detected using reverse engineering techniques.

c) What is software reengineering? State and explain any three conditions that will lead to the need for software reengineering required. [7 Marks]

It is re-organizing and modifying existing system to make them more maintainable. It involves:-

- Source code translation.
- Reverse engineering.
- Program structure development.
- Program modularization.
- Data re-engineering.

Restructuring or re-writing part or all of the legacy system without hanging its functionality.

When to Re-Engineer?

i) When the system changes are mostly confined to part of the system then re-engineer that part.

ii) When hardware or software support becomes obsolete.

iii) When tools to support re-structuring are available.

Question #5 [20 Marks]

a) State and briefly explain the three fundamental software configuration management activities. [6 Marks]

i) Version management – where support is provided to keep track of the different versions of software components. Version management systems include facilities to coordinate development by several programmers. They stop one developer overwriting code that has been submitted to the system by someone else.

ii) System integration – where support is provided to help developers define what versions of components are used to create each version of a system. This description is then used to build a system automatically by compiling and linking

iii) Problem tracking – where support is provided to allow users to report bugs and other problems, and to allow all developers to see who is working on these problems and when they are fixed.

b) Discuss any three issues to be considered for developing tactics for WebApp Configuration Management? [6 Marks]

i) Context

ii) People

iii) Scalability

c) Discuss the view that Open Source Software has decreased the productivity of developers and the quality of the systems produced. [8 Marks]

The answer to this section should:

- take the form of a discussion wherein evidence is presented in respect of clear productivity gains, limited improvements in quality (depending on one's definition of quality), open ended completion milestones.
- Productivity gains through the building of community of developers, and access to extensive and open sources of human expertise, and code libraries;
- Quality improvements using forums for continuous but informal structured walkthroughs, feedback, from many groups and individuals as "proof" readers or code inspectors;
- Deadlines and budgets are the most difficult to assess due to the nature of communities (primarily dependent on volunteers and interest groups), and the fact that in open source, development and maintenance are often fused together and difficult to separate.