INSTRUCTIONS TO STUDENTS

(1) This document consists of 5 pages including this cover page.

(2) This examination document consists of a total of seven (7) questions divided into two separate sections: Section A (4 questions) and Section B (3 questions).

(3) MANDATORY. Answer all 4 questions in Section A. Each question carries an equal score of 20 points. The total score for Section A is 80 points.

(4) SELECTION. Answer only 1 out of 3 questions in Section B. Each question carries an equal score of 20 points. The total score for Section B is 20 points.

(5) Please write/print your answers legibly in the Answer Booklet provided. In your answers, please write down the question reference number.
SECTION A

QUESTION A1 (20 points) – SOFTWARE PROJECT MANAGEMENT

You are working as a software engineer (SE) at WRY Associates Consulting. The firm recently announced that they have just secured a major contract for the development and production of a software application for a reputable client. For the new project, you are one of the firm’s candidates considered to be the Project Manager. An interview had been arranged to determine whether you are suited for the job.

During the interview the panel asked the following questions:
(Note: Please be concise, answer to address the questions being asked and not deviate to matters of irrelevance. Your answers will be graded on the quality of the ideas, its clarity and correctness).

A1.1 What do you do as soon as the project is handed down to you? (4 points)

ANSWER.

As soon as the project is handed down to a new Project Manager, the person must get to understand the full facts of the contract and study them closely before strategizing and planning the total project execution. Among the very important primary overview and relevant information regarding the project include:

- What is the project all about? The high level view, the scope of work and deliverables of the project.
- Who is the client (owner) of the project? What is the business case for the client to implement this project? Basically why the client wants this project.
- When is the project to commence? How long is the duration of this project? Is it a straight through implementation or a phased and staged out implementation?
- Where is the project going to be implemented (installation sites)? Local, remote and/or including the client’s third party sites.
- How much is the contract worth? What are the values of the various components (breakdown of major deliverables) of the contract worth?
- How was the contract awarded? Look for any formal documents for the project like Tender documents, Proposal Documents, Contract documents etc, including those legally binding and other supporting documents.
- Are there any major, special and significant ‘out of normal requirements’ in the contract that need to be satisfied?
- What is the nature of implementation of the contract? Single vendor, Joint-Venture, Contract terms – Design, Build, Operate and Transfer (DBOT, BOT etc), Consulting and Services only, Full Outsourcing, Services Outsourcing only, etc.

The main idea expected in the answer to this question is to ‘grasp’ the major preliminary, overview information pertinent to the project that forms the core data for the strategizing and planning of its implementation.
A1.2 List down 5 quality items of work in the project. How would you manage the quality of work for the items that you have specified? (4 points)

ANSWER

The 5 quality items of work (many more actually) related to software engineering practices complying with recognized and acceptable standards are listed below. To manage the quality of work for these 5 quality items, all we have to do is to ensure compliance to those standards.

(1) Quality in Project Management (ISO 9001) - Normally, a reputable software production company will have a quality system in place. If that is the case, the Project Manager will just have to follow and implement the project according to the existing quality system manual. If one does not yet exist, the Project Manager must put in place a workable quality assurance system to be implemented for managing the project. This item is about the total project management quality, of which the management of software quality is only one part of it. The Project Manager must at least consider complying with ISO 9001-2000 Quality Management System for this project, covering all the mandatory clauses in the various sections of the said standard.

(2) Standard for Software Project Management Plans (IEEE Std 1058) – In this case the coverage is specific for managing software projects, rather than generic projects like ISO 9001 which covers quality systems applicable to managing projects in all industries. For that reason, the clauses and sections in this IEEE Standard 1058 are also specific for the management of software projects.

(3) Software Engineering Product Quality Standard (ISO 9126) – In this case the clauses for compliance are focused on the quality of the software product itself. This standard defines that the 6 qualities for a software product that must be considered are: functionality, reliability, usability, efficiency, maintainability and portability.

(4) Standard for Software Quality Assurance Plans (IEEE 730 – 2002) – This standard focused on the planning and execution of documentation, testing, methodologies, tools, records, reviews, control, etc, during the project implementation to provide assurance of the quality of the software being manufactured.

(5) Standard for Software Life Cycle processes (ISO/IEC 12207) – This standard provides a full cycle at a high level structure of processes from conceptualization of the software to its retirement. It includes processes like user requirements, analysis modeling, software production, software testing, software configuration or change management, software quality management etc, to be conducted during the development of a software application until its retirement. It does not however, dictate a particular life-cycle model or software development method to be used.

A1.3 List down 5 anticipated risks in a project. How would you manage the anticipated risks that you have specified? (4 points)

ANSWER

The anticipated risks in a project can be viewed from many perspectives. One perspective is on: people, product, processes and quality. Another perspective is on: quality, processes, methods and tools. There are many risks in a project and the 5 anticipated risks we describe below are just to satisfy the instruction in this question.
(1) Quality – Risk on Not Meeting Requirements. If the user specifies some specific functional requirements and these requirements are not met in the final software product then the product is not usable. Under the quality criteria of usability, the software is not usable and people will not use it. This risk can be managed by ensuring that the exact user requirements are fully captured, documented and understood, and carried into the design, production and testing stages in the software development.

(2) People – Risk on Ineffective Decision Making. During the development of any project, one of the major problems normally reported is indecisiveness or untimely and delayed decisions for critical tasks (on the critical path) or major issues in the project. This can be costly in both monetary value and lost time. To mitigate this ineffective decision making, the project must be organized in a hierarchy such that the right people are given decision making powers, roles and responsibilities that are documented and stated clearly in the project charter and organization document.

(3) Methods – Risk on wrong Design Method. There can be situations that arise at the very end stages of the software development process like not meeting the criteria of complex queries to return results within a specific time period, like 2 seconds as an example. Looking back at it, the problem may seriously be due to the method used in the query algorithm and data gathering techniques in its design. This problem can be avoided by consulting experts with experience early on, and conducting early prototype tests on the query requirements before finalizing the software design. Some other techniques may possibly include parallel processing designs.

(4) Process – Risk on delays in activities. In cases where certain hardware that need to be procured and brought to site are known to take a long time (for example due to manufacturing and other trans-boundary approval and transportation requirements), the procurement process must be made very much earlier in time during the project implementation. The delays in these long lead items can cause overall delays to many parties and activities/tasks in the project plan. These delays can be avoided by early identification, careful planning, early execution and timely monitoring on the status of the procurement activity.

(5) Tools – Risk on Software Maintenance. Nowadays, there are many sophisticated CASE (Computer Aided Software Engineering) tools that can be used for software development. When many programmers work on a software project (most likely the case in a large complex software application), some standardization in coding styles, comments syntax, common coding methods, etc, must be enforced throughout the programming team. There must also be some means of enforcing version control (CVS – Concurrent Versioning System) for the various releases of components of the software application. Otherwise, the final software product could not be easily maintained. Choosing the right CASE tool and enforcing strictly to agreed programming standards will avoid these problems.

A1.4 Explain how you would manage the costs in the project. (4 points)

ANSWER

There are many ways of managing costs in a project. From the point of the implementation schedule, ensure that assigned tasks are completed on time or better because time is money. From the point of view of purchasing and procurement, go and renegotiate the basic costs with the suppliers and vendors, without compromising quality. From the point of view of transportation (if there are a lot of purchase items), organize them such that the delivery can
be synchronized and lumped in bigger parcels. It is certainly a good idea to hire a shrewd project accountant and an experienced procurement officer to be in the project team.

A1.5 Explain how you would manage the deliverables of the project. (4 points)

ANSWER

The set of deliverables for a project is dependent the contract document and scope of work specified therein. First and foremost, we need to identify these deliverables which can vary from hardware, systems (software applications) or services (consulting, programming, training, auditing) etc. Second, we have to plan, document and schedule the delivery of these deliverable items. Third, we have to constantly monitor the progress of work and intervene when anticipated problems and delays may arise. Fourth, we have to provide appropriate corrective actions and measures where and when required. Fifth, we hope that decisions that need to be made are made timely so as not to jeopardize the overall implementation of the project. Lastly, pray that no untoward incidences occur like "Actions of God" that cannot be prevented during the course of the project. In some cultures, for major projects there is always some prayer sessions (like slaughtering of animals for feasts or offerings) to seek the blessings of the "Majestic Power Up There".
QUESTION A2 (20 points) - SOFTWARE QUALITY AND TESTING

A2.1 The ISO 9126 Software Engineering Product Quality standard defines specific terms for evaluating software quality. These are classified under 6 categories of the so called “ilities”. List these 6 categories. (6 points)

ANSWER

(1) Functionality – the required functions must be available in the software.

(2) Reliability – stable in use, can maintain performance in stated conditions

(3) Usability – be assessed by users as usable and easy to use

(4) Efficiency – be efficient in resource and response performance in stated conditions

(5) Maintainability – be assessed as easily modifiable and maintained

(6) Portability – be easily transferable from one software platform to another

A2.2 ISO 9126 distinguishes between a defect and a non-conformity. Explain the difference using appropriate examples. (2 points)

ANSWER

A non-conformity is a non-fulfillment of a specified requirement. For example, if a query function was documented as a specific requirement in the User Requirement Specifications (URS), and the software does not have that query, then the omission is called a non-conformity.

A defect is a non-fulfillment of an intended usage requirement. For example, if the required query as stated above upon execution fails to produce the right results, then the failure is a defect.

Basically “defect” is defective against intended use, whereas “non-conformity” is not conforming or not complying with some specific documented requirement.

Another way of stating it in metaphors:
“A non-conformity is not doing the right thing and a defect is not doing the thing right”.

A2.3 In software testing, a distinction is made between validation and verification. Explain the difference using appropriate examples. (2 points)

ANSWER

Validation (for software): The process of evaluating software to ensure compliance with specified requirements. (ISO 9000-3: 1991, 3.7) This answers the question “Have we built the right product?”

Verification (for software): The process of evaluating the products of a given phase to ensure correctness and consistency with respect to the products and standards
provided as input to that phase. (ISO 9000-3: 1991, 3.6) Simply stated, for a given set of inputs to the software, upon testing we should get the expected set of correct outputs. This type of test verifies that the software has performed correctly. So verification answers the question “Have we built the product right?”

A2.4 Explain what is meant by the term “white-box” testing. (2 points)

ANSWER

In software testing, if you have a module for example and you tested every possible path in the module. You call this “white-box” testing.

It means for white-box testing, you test from the inside ('white-box' or everything inside is known to you), so it is achievable to test all possible paths. In some cases, it may not be practical to conduct a hundred percent white-box test.

A2.5 Explain what is meant by the term “black-box” testing. (2 points)

ANSWER

In software testing, if you verify the 'documented' or 'stated' functionality of the module or the complete system, but without testing every possible path of the module or of the complete system, then the testing is called “black-box” testing.

It means for black-box testing, you test the documented functionality from the outside ('black-box' or inside is unknown to you), whether you are testing just the module or the full complete system.

A2.6 Describe the following terms in the context of practice in software engineering and software quality testing. (6 points)

ANSWER

- **Unit Test**
  The test for verification on the smallest unit of software design as implemented in source code, i.e. the software component or module.

- **Integration Test**
  The systematic technique of conducting tests on the software architecture as it is being built by unit components so as to uncover errors associated with component interfacing.

- **Sub-system Test**
  The sub-system test is the system test for a sub-system when a large system is broken down to smaller pieces.

- **System Test**
  The test on a system comprising all the elements of hardware, software, network etc, to fully exercise the system in operation to ensure that every element have been properly integrated and performing the required functions as designed.
• Performance Test
  The test for the run-time performance of the integrated software for response performance measured against demands of resource utilization like CPU cycles and memory.

• Acceptance Test
  The test of the full system as in the intended operation: with users actually running the various subsystems and functionality, putting every conceivable usage into action, etc, covering all the stated user requirements before acceptance by the client.
QUESTION A3 (20 points) - REQUIREMENT ANALYSIS & DESIGN

You applied for a job and are now being interviewed for the position of Software Engineer in a software production house. A major part of your job involves gathering user requirements and communicating those requirements to the Software Designers in the company.

In order to gauge your level of understanding for the job, the interview panel decided to show you an example of a typical communication document within the company. You were then asked to answer a few questions related to the document.

The following questions from the interview panel refer to the figure above.

A3.1 What type of UML diagram is this? In software design, what information does this selection of UML diagram convey? (2 points)

ANSWER

The diagram is a UML class diagram. It displays the static class objects and the relationships among the objects in the software system. The class diagram conveys information like the name of objects, the properties or data (some also provide data-types) for the object instances, the operations or methods (later coded as procedures, subroutines) for each object instance, the relationship or linkage of one object instance into another (one-to-one, one-to-many, etc), the subclass and superclass (inheritance relationship) etc.

A3.2 In the context of software design and modeling, explain the meaning and use of the terms below:

ANSWER

- Association (2 points)
An association is a link connecting two classes. The association exists if an instance of one class must know about the other in order to perform its work.

- **Abstract class (2 points)**
  A class that cannot be instantiated is an abstract class. It is written in italics and bold. (A normal class is just shown bold) The abstract class provides properties and methods that are automatically inherited by its subclasses.

- **Generalization (2 points)**
  A generalization is an inheritance link indicating one class is a superclass of the other. The class with the arrowhead is the superclass (general) and the one without is the subclass (with specific details). The subclass inherits everything in the superclass.

- **Navigability (2 points)**
  The navigability is the arrowhead on an association which shows the direction the association can be traversed or queried.

**A3.3** Based on the information in the figure above, you are to ‘reverse’ the situation and list down (sentences in point form as we have conducted in class) the 'User Requirement Specifications' that led to the production of this UML diagram. (10 points)

**ANSWER**

A typical set of user requirement that translates into the above class diagram is described below:

- A customer detail consists of a name and an address.
- A customer can make many orders or no orders at all.
- An order information consists of the date or order, status of order, the order amount and a payment method which can be only one of the three methods - by credit, by cash, by check.
- Authorization is required if payment is made by credit or by check. No authorization is required for cash payments.
- For check payments, the name of the bank and the bank_id information is required.
- For credit payments, the credit card type, the card number and card expiry date are required.
- Each order must have at least one line item (OrderDetail). An order can have a collection of line items (aggregate of line items).
- Each line item must have a quantity and a tax status.
- An item may not be in any line item (OrderDetail). An item may be in many line items (OrderDetail).
- Each item must have an associated description and a shipping weight.
QUESTION A4 (20 points) - SOFTWARE DESIGN SPECIFICATIONS

In software engineering practice, the design of software is normally conducted by separating the documented specifications into two components: the Description Specification and the Operational/Functional Specification.

A4.1 Explain what is meant by the Description Specification. (2 points)

ANSWER

The description specification is the description of the overall system architecture, the software environment and platform, the application layers, static data perspectives (both textual and graphical) and their relationships like the structure of packages, package diagrams, classes and objects, class diagrams, component and deployment diagrams, etc to meet the informational requirements of the software.

A4.2 Explain what is meant by the Operational/Functional Specification. (2 points)

ANSWER

The operational or functional specification is the description (both textual and graphical) information that focuses on the behavior or operations of the software components, their relationships, communications and messaging to meet the functionality requirements. Examples of these descriptions include use-cases, collaboration diagrams (with messages) state diagrams (with state changes), sequence diagrams and activity diagrams. The perspective of this specification is on the object relationships with regards to software functionality and their dynamic changes.

A4.3 By analyzing the model diagram shown below, explain what you can make up from the information displayed and put into the write-up of the Software Design Specification. (8 points)
The diagram above is a state (statechart) diagram. The states are drawn in rounded boxes instead of the sharp cornered boxes for classes. The diagram displays all possible states of the object and the transitions that cause the change of state. The state of an object depends on its current activity or condition. Transitions are arrows from one state to another. Events or conditions that trigger transitions are written beside the arrows. These triggers are external transitions between one state and another.

The diagram shows two self-transitions for the “Getting SSN” and “Getting PIN” states shown through the respective self-looping arrows for each state. This is triggered by pressing keys or their combinations.

The action that occurs as a result of an event or condition is expressed in the form ‘action’. In the case of the “Validating” state in the diagram above, while in its state, the object does not wait for an outside event to trigger a transition. Instead, it performs an activity. The result of that activity determines its subsequent state.

The diagram shows one initial state (a dummy state to start the action) and various possible final states (dummy states showing the end of actions).

A4.3 Similarly, based on the model diagram shown below; explain what you can make up from the information displayed and put into the write-up of the Software Design Specification. (8 points)

The diagram above is an entity-relationship diagram (ERD). The primary purpose of the ERD diagram is to represent data objects and their relationships. Usually it is used for the design and specification of relational database tables for the software system. It is important to note that the ERD diagram is not meant to display the behavior, messaging and communications aspects of the various software components.
The above ERD diagram is about the entity "Invoice". The six solid boxes (MSF 260, MSF 26A, MSF 270, MSF 261, MSF 262 and MSF263) are precisely the primary data objects for this entity and the ERD diagram shows its relationships to each other. Normally each data object is translated to a specific database table. The lines connecting the data objects specify the multiplicity of data relationships like one-to-one, one-to-many, many-to-many etc between the records in each database table. These record linking relationships are implemented in the database tables through primary and secondary keys.

The five dotted boxes (Module 3301, Module 3280, Module 3380, Module 3210 and Module 3901) show secondary relationships of "Invoice" data to the various data objects in the various modules inside the overall software system.
**SECTION B (Answer only 1 out of the 3 questions below)**

**QUESTION B1 (20 points) – USER REQUIREMENTS STUDY**

Requirements engineering is the technical term (fancy word) basically for the activities of "User Requirements Study (URS), Analysis and Documentation". This activity is the precursor to the activities that led to documenting the "Software Requirements Specifications (SRS)".

**B1.1 Explain by giving five specific examples why it is crucial to have very high quality information in the “User Requirement Study” document. (10 points)**

**ANSWER**

The “high quality” of information in the URS document serves two purposes: One specifies exactly the requirements of the user in the language the user understands, the other is the requirements for the developer to ensure that the developer understands what the user wants.

Usually, the URS document contains both technical and non-technical information expressed in a way such that the user understands and can refer to. That is why there are such things like GLOSSARY, ABBREVIATIONS, DEFINITIONS, and so on.

The quality of information must at least cover statements and descriptions for "the what", "the when", "the where", "the who", "the how", "the how much", "the how many", etc, regarding the requirements. It is very important for the following reasons:

1. to avoid misunderstanding between the expectations of the user and the developer
2. to provide information to the user on the specific ‘receivables’ of the project
3. to provide information to the developer on the specific ‘deliverables’ of the project
4. to provide the developer all the necessary information as to the total scope of work, to make the plans, schedules, resources, to design the software, to execute the implementation, to assign various responsibilities to both the user and the developer, to produce other important documents along the implementation to be agreed by both the user and the developer. It is some sort of a "legally binding contract‘ mutually agreed by the two parties.
5. All the above reasons (a) to (d) help to ‘smoothen’ the implementation and basically make both parties happy. Both are interested parties and work together as partners in the project. Everybody gets what they wanted and nobody will go to court to sue each other.

**B1.2 The IEEE Computer Society provides a standards document IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830 – 1998). List down and describe the five main sections (clauses) that must be covered in the SRS. (10 points)**

**ANSWER**

(1) Overall structure of the SRS document to contain sections: Purpose, Scope, Definitions, acronyms and abbreviations, References, Overview, etc.
(2) Overview description of the software product: Product perspective, Product functions, User characteristics, Constraints – inclusion and exclusion of specifics in the scope of the product, Assumptions and dependencies of the product to other entities, etc.

(3) Specific functionality requirements – what the user wants in the software application including functionality and performance.

(4) Specific infrastructure requirements: hardware, software platforms, integration with existing systems, network and communications, preferences to specific platforms, etc.

Microsoft Project is one of the many software tools that have successfully been used in managing major projects. As a software engineer, you will most likely be working with a project team that uses some kind of project management tool and you are required to know at least the basics of how to use this tool.

The software development company you are currently working with have just assigned you on probation to one of the firm’s software development teams. Before sending you to meet the Project Manager, you were asked the question: “When do you use the following charts and what types of project information are available and displayed directly in those charts?”

B2.1 Gantt Chart (4 points)

ANSWER

The Gantt Chart view displays task information about your project as both text and bar graphics. On the left side of this view are columns (containing Microsoft Project fields) in which you can enter and modify task names, durations, start and finish dates, and other information. On the right side of this view, you can use Gantt bars to graphically display task durations and start and finish dates on a timescale. The relative position of the Gantt bars shows the sequence in which your project tasks are scheduled to occur.

B2.2 Network Chart (4 points)

ANSWER

The Network Chart (Diagram) view displays tasks and task dependencies in a network or flowchart format. A box (sometimes called a node) represents each task, and a line connecting two boxes represents the dependency between the two tasks. By default, the Network Diagram view displays one diagonal line through a task that is in progress and crossed diagonal lines through a completed task.

B2.3 Task Usage (4 points)

ANSWER

The Task Usage view displays project tasks with their assigned resources grouped underneath them. The information available in the Task Usage view relates to its usage like activities to:

- Assign people and other resources to tasks.
- Enter and edit task and resource information together, such as work, start and finish dates, cost, work allocation, and work availability.
- Distribute task assignments more evenly across resources.
- Find out how many hours each resource is scheduled to work on particular tasks.
• Vary the amount of work a person spends on a task by setting work contours.
• Split a task so that the second portion of it starts at a later date.

B2.4 Resource Sheet View (4 points)

ANSWER

The Resource Sheet view displays information about each work or material resource (such as the payment rate, the number of work hours assigned, and the baseline and actual cost) in a spreadsheet-like format. You can use the Resource Sheet view to:

• Enter and edit work or material resource information.
• Review the number of hours of work assigned to each resource.
• Review resource costs.

B2.5 Resource Usage View (4 points)

ANSWER

The Resource Usage view displays project resources with their assigned tasks grouped underneath them. The information available in the Resource Usage view relates to its usage like activities to:

• Enter and edit information on a resource's task assignment, such as cost, work allocation, and work availability.
• See which resources are over-allocated and by how much.
• Distribute assignments more evenly among resources.
• Find out how many hours each resource is scheduled to work.
• See the percentage of capacity at which each resource is scheduled.
• Determine how much time each resource has available for additional work assignments.
• Find out how many hours each resource is scheduled to work on particular tasks.
• Review resource costs on a particular task.
• Vary the amount of work a person spends on a task by setting work contours.
**QUESTION B3 (20 points) – SOFTWARE MAINTENANCE & CONTROL**

Change Management, more commonly called Software Configuration Management (SCM) is a very important activity that is applied throughout the software engineering process. For that reason, this subject is addressed fully in the IEEE Standard for Software Configuration Management Plans (IEEE Std 828 – 1998).

---

**B3.1** What are the five categories (classes) of information that you need to document to build up the Software Configuration Management Plan, SCMP? What are the uses of each of this information (10 points)

**ANSWER**

1. SCM Management (Who): Use to identify the responsibilities and authorities for accomplishing the planned activities for any changes in the project.

2. SCM Activities (What): Use to identify all activities to be performed in applying any changes to the project.

3. SCM Schedules (When): Use to identify the required coordination of SCM Activities with the other activities in the project.

4. SCM Resources (How): Use to identify tools, physical and human resources required for the execution of the plan.

5. SCM Plan Maintenance (Control): Use to identify and control how the SCM Plan will be kept current while in effect.

---

**B3.2** List down five of the many configuration items (CI)/information that you need to identify in a software project for the SCMP. (5 points)

**ANSWER**

The five examples of controlled configuration items (CI) in the Software Configuration Management Plan are listed below:

1. Executable codes.

2. Source codes.

3. Program listing.

4. Database information.

5. Test cases.
B3.3 Based on the five configuration items that you have specified in question B3.2 above, describe by using examples on how you plan to execute their configuration control. (5 points)

ANSWER

(1) Executable codes – We control the executable codes by maintaining the list of executable codes, names of executables, details, date last modification, person who last modified the codes, and locations of these executables in the system to facilitate changes and replacements.

(2) Source codes – We control the source codes by maintaining the list of the source codes, source code files, revision numbers, comments or descriptions on change actions history, authors, etc to facilitate changes and replacements.

(3) Program listing – We control the programs and its listing by maintaining the list of programs, its locations, its configuration settings, its inter-relationship information with dependent programs to facilitate changes and replacements.

(4) Database information – We control the database changes by maintaining the full database information (structural data relationships, tables, indexes, sequences, saved queries, saved procedures and functions, saved triggers, archive tables, logging information, etc) to ensure that future persistent data requirements and changes can be easily accommodated.

(5) Test cases – We control the testing and test cases by maintaining the recorded test cases and their expected results to be used by new tests or incorporated into newly created tests when changes and modifications are made to any component in the software system.

(Author: WRY wruslan@mmu.edu.my)