SLIATE

SRI LANKA INSTITUTE OF ADVANCED TECHNOLOGICAL EDUCATION

(Established in the Ministry of Higher Education, vide in Act No. 29 of 1995)

Higher National Diploma in Information Technology

Second Year Second Semester Examination – 2022

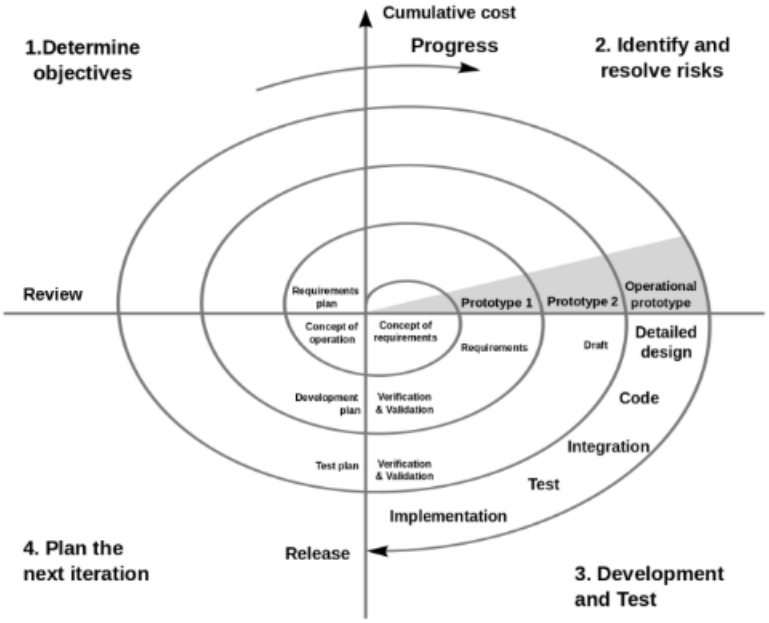
HNDIT 4012-Software Engineering

Marking Scheme

Q1)	
I	**04 marks**
Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.	
II	**06 marks**
Generic products Stand-alone systems that are marketed and sold to any customer who wishes to buy them. Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists. **Customized products** Software that is commissioned by a specific customer to meet their own needs. Examples – embedded control systems, air traffic control software, traffic monitoring systems.	03 marks 03 marks
III	**04 marks**
Stand-alone applications Interactive transaction-based applications Embedded control systems Batch processing systems Entertainment systems Systems for modelling and simulation Data collection systems Systems of systems	Any 04 answers

IV	06 marks
Confidentiality- Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed. **Competence** Engineers should not misrepresent their level of competence. They should not knowingly accept work which is out with their competence. **Intellectual property rights** Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. **Computer misuse** Software engineers should not use their technical skills to misuse other people's computers. **Business and social change** Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. **Security and trust** As software is intertwined with all aspects of our lives, it is essential that we can trust that software.	Any 03 answers
	Total 20 marks

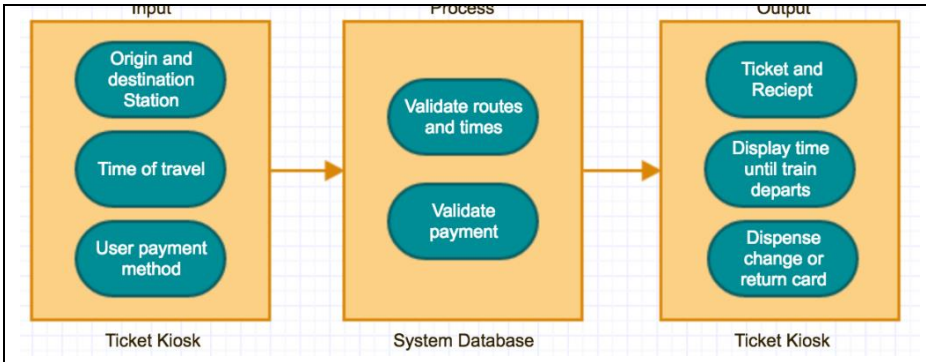
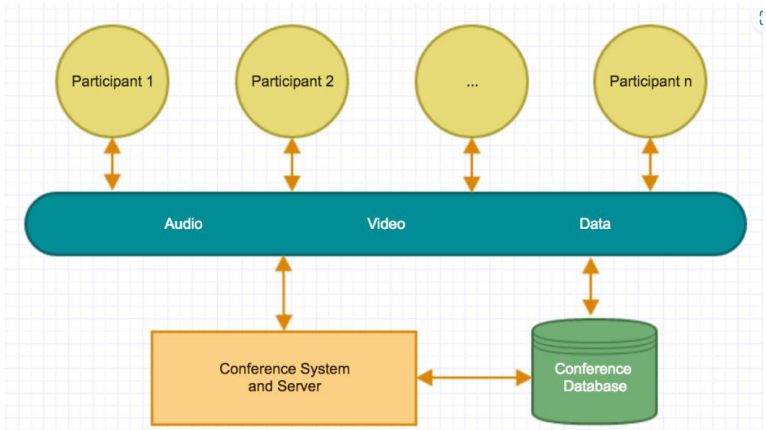
Q2)	
I	**04 marks**
A structured set of activities required to develop a software system. Specification Design and implementation Validation Evolution	02 marks 02 marks
II	**06 marks**
Waterfall model Clarity and Simplicity Clearly Defined Phases Sequential Approach Plan driven Stability in Requirements **Prototype model** Quick client feedback Developed prototypes can be used later for any similar projects. Any missing functionality and any error can be detected early. It is useful when requirements are not clear from the client's end.	03 marks 03 marks

III	04 marks
	
IV	**06 marks**
Extreme Programming (XP) is a software development methodology that focuses on delivering high-quality software quickly and efficiently. Throughout the process, the Scrum Master facilitates the Scrum events, removes impediments, and coaches the team on Scrum practices. The Product Owner collaborates with stakeholders to ensure that the product backlog reflects their needs and priorities. A kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow). It include todo, doing and done works.	02 marks 02 marks 02 marks
	Total 20 marks

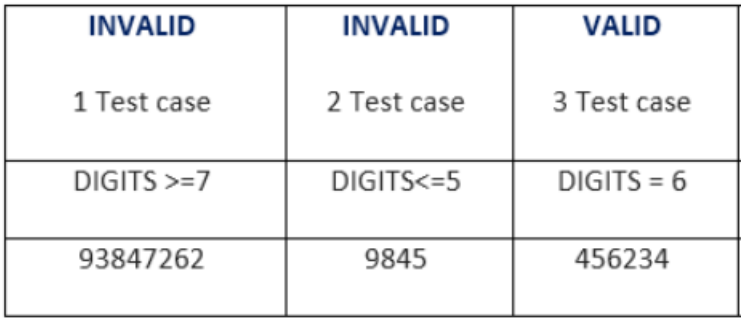
Q3)	
I	**04 marks**
Software analysis includes all activities, which help the transformation of requirement specification into implementation. The collected data is analyzed to conceptualize what needs to be done. This involves identifying functional and non-functional requirements. Software specification provides a detailed description of the requirements that need to be fulfilled for the successful development of a software system.	

II	**06 marks**
Functional requirements describe what the software system must do. They specify the system's behavior, features, and interactions with users and other components. Non-functional requirements define how the system should perform rather than what it should do. They cover aspects like performance, security, usability, and reliability. Constraints are limitations or restrictions that affect the design, development, or deployment of the system.	02 marks 02 marks 02 marks
III	**04 marks**
Sampling: This involves analyzing a subset of the target user base or system stakeholders. Research: Conducting research involves studying existing documentation, reports, and relevant literature. Observation: Directly observing users or stakeholders in their natural environment can reveal valuable information. Questionnaires: Surveys and questionnaires allow you to collect structured data from a larger group of people. Interviews: Interviews involve one-on-one conversations with stakeholders. T Prototyping: Creating a prototype (a simplified version of the system) allows stakeholders to interact with it and provide feedback.	Any 04 answer
IV	**06 marks**
Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. Structure chart represents hierarchical structure of modules. At each layer a specific task is performed. Pseudo code is written moreclose to programming language. It may be considered as augmented programming language, full of comments and descriptions. A Decision table represents conditions and the respective actions to be taken to address them, in a structured tabular format. Data dictionary is the centralized collection of information about data. It stores meaning and origin of data, its relationship with other data, data format for usage etc. So, all the things required to develop quality projects. UML Any other possible answer	Any 03 answers
	Total 20 marks

Q4)	
I	**04 marks**
Modularization is a technique to divide a software system into multiple discrete and independent modules, which are expected to be capable of carrying out task(s) independently. Advantages • Smaller components are easier to maintain. • Program can be divided based on functional aspects. • Desired level of abstraction can be brought in the program • Components with high cohesion can be re-used again. • Concurrent execution can be made possible. • Desired from security aspect. (Any two)	02 marks 02 marks
II	**06 marks**
Cohesion refers to the degree to which elements within a module work together to fulfill a single, well-defined purpose. In other words, it measures how closely related the elements (such as functions, classes, or components) are within a module. High cohesion indicates that these elements are closely related and contribute collectively to a specific functionality. Low coupling refers to the idea that each module or component of your code should be as independent as possible from other modules or components. In other words, changes to one component should not require changes to other components.	03 marks 03 marks
III	**04 marks**
Software architecture refers to the high-level structure or organization of a software system, which defines the components, relationships, and interactions that make up the system. MVC Layered Client Server Repository Pipe and Filter (Any two)	02 marks 02 marks
IV	**06 marks**
Ticket Machine Pipe and filter Layered	 01 mark

	02 mark
Video-conference system Pattern client server	01 marks
	02 marks
	Total 20 marks

Q5)	
I	**04 marks**
Code walk through Code walk through is an informal code analysis technique. The main objectives of the walk through are to discover the algorithmic and logical errors in the code. A few members of the development team are given the code few days before the walk-through meeting to read and understand code. **Code inspection** The aim of code inspection is to discover some common types of errors caused due to oversight and improper programming.	02 marks 02 marks
II	**06 marks**
Rules for limiting the use of global. Naming conventions for global variables, local variables, and constant identifiers. Contents of the headers preceding codes for different modules. Comments	Any 03 answer

Do not use a coding style that is too clever or too difficult to understand. The code should be well-documented.	
III	**04 marks**
Black-box testing In the black-box testing approach, test cases are designed using only the functional specification of the software, i.e. without any knowledge of the internal structure of the software. Equivalence class portioning Boundary value analysis **White-box testing** white-box testing approach, designing test cases requires thorough knowledge about the internal structure of software, and therefore the white-box testing is called structural testing. Statement Coverage Decision Coverage Branch Coverage	02 marks 02 marks
IV	**06 marks**
	
	Total 20 marks

Q6)	
I	**04 marks**
Software quality refers to the overall goodness or excellence of a software product. It encompasses various attributes that determine the software's reliability, maintainability, performance, and user satisfaction. Functional Correctness Reliability Usability	
II	**06 marks**
ISO 9001 certification is a globally recognized standard for quality management. It helps organizations improve their performance, meet customer expectations, and demonstrate their commitment to quality. Preparation Documentation Implementation	03 marks

Internal audit Certification SEI CMM can be used two ways: capability evaluation and software process assessment. Capability evaluation and software process assessment differ in motivation, objective, and the final use of the result. Capability evaluation provides a way to assess the software process capability of an organization.	03 marks
III	**04 marks**
Corrective Maintenance - This includes modifications and updating done in order to correct or fix problems, which are either discovered by user or concluded by user error reports. Adaptive Maintenance - This includes modifications and updating applied to keep the software product up-to date and tuned to the ever-changing world of technology and business environment.	02 marks 02 marks
IV	**06 marks**
Agree The standard age of any software is considered up to 10 to 15 years. Older software, which were meant to work on slow machines with less memory and storage capacity cannot keep themselves challenging against newly coming enhanced software on modern hardware. As technology advances, it becomes costly to maintain old software. Most maintenance engineers are newbie and use trial and error method to rectify problem. Often, changes made can easily hurt the original structure of the software, making it hard for any subsequent changes. Changes are often left undocumented which may cause more conflicts in future.	01 mark Any suitable answer 05 marks
	Total 20 marks

End of the Marking Scheme