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, First Semester Examination – 2016

HNDIT2312- Principles of Software Engineering

Instructions for Candidates: Answer four (4) questions only

No. of questions : 05
No. of pages : 03
Time : Two (2) hours

1.

i. What it meant software engineering? [03 Marks]

Software engineering is an engineering discipline which 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. In this definition there are two key phrases

ii. Explain the difference between software engineering and system engineering [04 Marks]

System engineering is concerned with all aspects of computer-based system development, including hardware, software and processor engineering. Software engineering is a part of this process

iii. Write four reasons for failure of software product. [04 Marks]

Time Schedules and cost estimates of many software projects are grossly inaccurate.
Software is costly.
The quality of software is not satisfactory
Software is difficult to maintain.
The productivity of software people is not satisfactory to meet the demand.

iv. Briefly explain four (04) software process characteristics. [08 Marks]

Understandability →To what extent is the process explicitly defined and how easy is it to understand the process definition?

Visibility → Do the process activities culminate in clear results so that the progress of the process is externally visible?

Acceptability → Is the defined process acceptable to and usable by the engineers responsible for producing the software project?
Reliability → Is the process is designed in such a way that process errors are avoided or trapped before they result in product errors?
Rapidity → How fast can the process of delivering a system from a given specification be completed?
Robustness →Can the process continue in spite of unexpected problems?
Maintainability →Can the process evolve to reflect changing organizational requirements or identified process improvements?
Supportability →To what extent can the process activities be supported by CASE tools?

v. You have to develop high risk software project. What is the process model you select for developing the above project?
Spiral model
[02 Marks]

vi. What are the four (04) main task regions in spiral model? [04 Marks]
Determine goals, alternatives, and constraints
Evaluate alternatives and risks
Develop and test
Plan

(Total 25 Marks)

2.

i. What is the deliverable at the end of the process in requirements specification? [03 Marks]
A detailed and precise description of the system requirements is set out or SRS document

ii. What is requirement validation? [03 Marks]
Requirements validation is the process of checking that requirements actually define the system that the customer really wants.

iii. What is requirement elicitation and analysis? [04 Marks]
In this activity, technical software development staff work with customers and system end-users to find out about the application domain, what services the system should provide, the required performance of the system, hardware constraints and so on.

iv. Name of the all techniques used in the requirement validation process. [04 Marks]

Requirements Reviews
Prototyping
Test-case generation
Automated consistency analysis

v. Different types of checks should be carried out on the requirements validation process. They must be included in the requirements document also. List all of them. [05 Marks]

Validity checks
Consistency checks
Completeness checks
Realism check

Verifiability

vi. The requirements document is an official statement of what the software system developers worked on. What should include in the requirements document and name any two readers for each?
[06 Marks]
User requirements (Requirements definition)

Any two
Client managers
System end-users
Client engineers
Contractor managers
System architects

System requirements

Any two
System end-users
Client engineers
System architects
Software developers
Software Requirements Specification (SRS)

System architects
Software developers

(Total 25 Marks)

3.

i. Briefly explain the following software design principles.

a. Encapsulation /Information hiding [02 Marks]

Encapsulation is a technique for minimizing interdependencies among separately written modules by defining strict external interfaces.
The external interface acts as a contract between a module and its clients.
If clients only depend on the interface, modules can be re-implemented without affecting the client. Thus the effects of changes can be confined.

b. Modularity [02 Marks]

Software is divided into separately named, addressable components called modules. Complexity of a program depends on modularity. Modularity facilitates for development process, maintenance process, project management process and reusability

ii. What is meant by software architectural design? [02 marks]

The initial design process of identifying these sub-systems and establishing a framework for sub-system control and communication is called architectural design

iii. Write down two (02) benefits of software reuse. [04 marks]

Increased reliability, Reduced process risks, Effective use of specialists, Standards compliance, Accelerated Development

iv. Briefly explain repository model.

Keeping all shared data in a central database that can be accessed by all sub-systems. A system model based on a shared database is called *repository model*

a. List two (02) advantages of repository model [02 marks]

no need to transmit data explicitly from one sub-system to another, Activities such as backup recovery, access control and recovery from error are centralized, The model of sharing is visible through the repository schema

b. List two (02) disadvantages of repository model [02 marks]

Sub-systems must agree on a repository data model. This is a compromise between the specific needs of each tool, difficult or impossible to integrate new sub-systems if their data models do not fit the agreed schema, Evolution may be difficult as a large volume of information is generated according to an agreed data model. Translating this to a new model will certainly be expensive

v. Compare and contrast Cohesion & Coupling is in software design. [08 marks]

Cohesion	Coupling
Cohesion is the indication of the relationship within module.	Coupling is the indication of the relationships between modules.
Cohesion shows the module's relative functional strength.	Coupling shows the relative independence among the modules.
Cohesion is a degree (quality) to which a component / module focuses on the single thing.	Coupling is a degree to which a component / module is connected to the other modules.
While designing you should strive for high cohesion i.e. a cohesive component/ module focus on a single task with little interaction with other modules of the system.	While designing you should strive for low coupling i.e. dependency between modules should be less.
Cohesion is the kind of natural extension of data hiding for example, **class** having all members visible with a package having default visibility.	Making private fields, private methods and non public classes provides loose coupling.

vi. List three (03) software design techniques. [03 marks]

Top-Down Decomposition, Bottom –Up Design, Jackson Structured Design, Object Oriented Design (any 03)

(Total 25 Marks)

4.

i. Briefly explain following configuration management concepts? [04 Marks]

a. Release management

Releases must incorporate changes forced on the system by errors discovered by users and by hardware changes

They must also incorporate new system functionality

Release planning is concerned with when to issue a system version as a release

(Any answer, 01 mark)

b. Configuration management planning

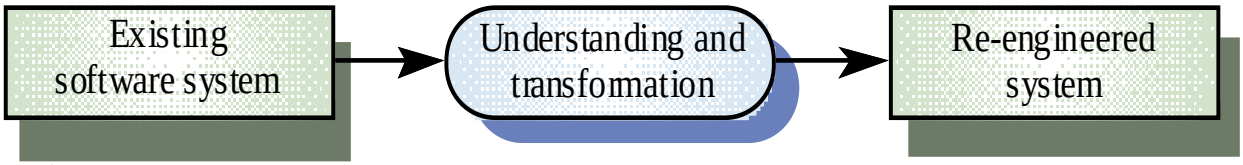
The purpose of this plan is to provide basic guidelines for how source code and software builds will be managed.

This document will cover source code administration, build environment standards and define the process by which new components will be added to builds, and a common understanding of how to manage broken builds.

(Any answer, 01 mark)

ii. Briefly explain Software forward-engineering and Re-engineering. [06 Marks]

Forward engineering



Software re-engineering

iii. Briefly explain about project scheduling activities during the software life cycle? [05 Marks]

- Split project into tasks and estimate time and resources required to complete each task.
- Organize tasks concurrently to make optimal use of workforce.
- Minimize task dependencies to avoid delays caused by one task waiting for another to complete.
- Dependent on project manager's intuition and experience
- Use grant chart and activity diagrams

(Any 03 answers, 05 marks)

iv. Why critical path is important in software projects? [02 Marks]

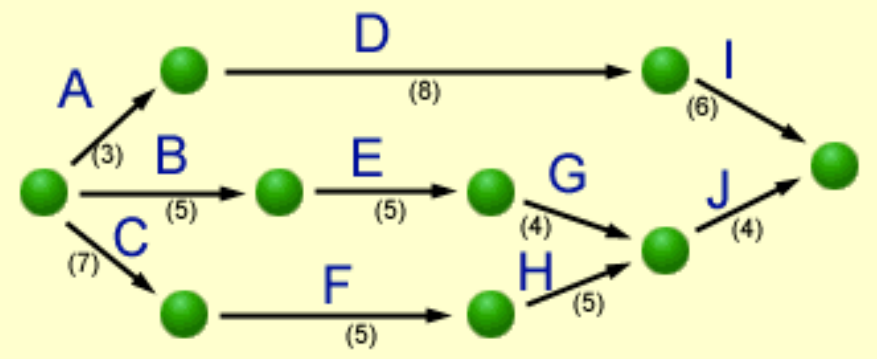
CPM is a project network analysis technique used to predict total project duration
A critical path for a project is the series of activities that determines the earliest time by which the project can be completed
The critical path is the longest path through the network diagram and has the least amount of slack or float.
If one or more activities on the critical path take longer than planned, the whole project schedule will slip unless corrective action is taken
(Any 02 answers, 05 marks

v. The following table explains the activities, duration and dependencies

Task	Predecessors Tasks (Dependencies)	Time (Weeks)
A	-	3
B	-	5
C	-	7
D	A	8

E	B	5
F	C	5
G	E	4
H	F	5
I	D	6
J	G - H	4

a. Draw the Activity on Arrow diagram? [05 Marks]



b. find the critical path [03 Marks]

The critical path is through activities C, F, H, J

The expected project duration is 21 weeks (7+5+5+4)

(Total 25 Marks)

5.

i. Why do we need to do software maintenance? [04 marks]

Errors in the existing system

Changes in requirements

Technological advances

Legislation and other changes

ii. List three types of software maintenance and briefly explain them [06 marks]

Corrective Maintenance

a. to repair software faults(fixing bugs in the code)

Adaptive Maintenance (adapting the software to new environments)

b. Maintenance to add to or modify the system's functionality

c. Modifying the system to satisfy new requirements

d. Modifying the system to suit new operation environment

Perfective Maintenance

e. Improving programs performance, structure, reliability etc. Making changes to avoid future problems or prepare for future changes

iii. What is the difference between Validations and Verification in software testing? [02 marks]

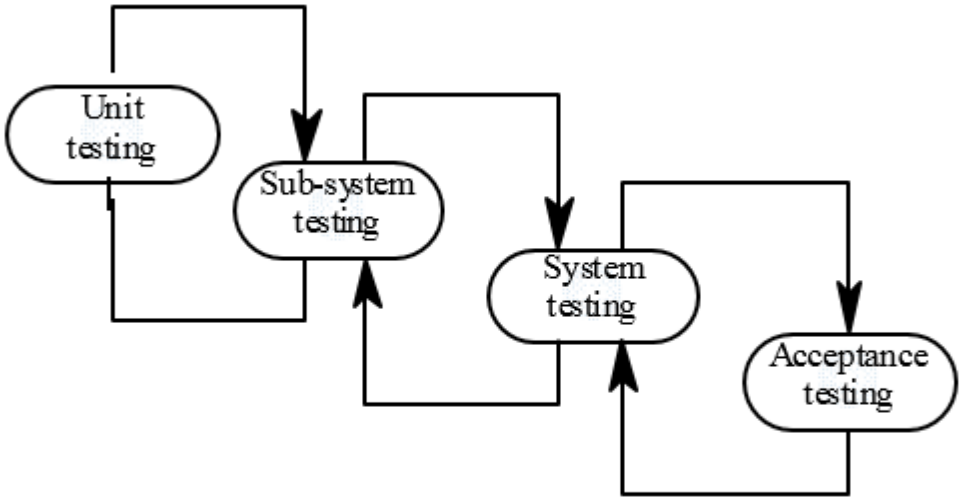
Validation and verification (V & V) is the name given to the checking and analysis processes that ensure that software conforms to its specification and meets the needs of the customers who are paying for the software.

V & V is a whole life-cycle process. It starts with requirements reviews and continues through design reviews and code inspections to product testing. There should be V& V activities at each stage of software process.

Validation : Are we building the right product?

Verification : Are we building the product right?

iv. Briefly explain the dynamic software testing process with a diagram. [03 marks]



v. List 05 types of software testing methods and briefly explain two. [06 marks]

Black-box testing
White-box testing
Gray-box testing
Unit Testing
Integration/Sub-systems Testing
Incremental Integration Testing
Alpha Testing

Beta testing

vi. Briefly explain following Static verification techniques . [04 marks]

Code walkthroughs – The author explain the code to the other team members. They examine the code and suggest improvements.

Code reviews (program inspections) – The author distribute the code lists to the team members. At a review meeting improvements are suggested

(Total 25 Marks)