

Model answers

Higher National Diploma in Information Technology

Second year, First Semester Examination – 2017

HNDIT2311- Principles of Software Engineering

Instructions for Candidates:
Answer only four (04) Questions

No. of questions: 05
No. of pages : 03
Time : 02 hours

1.			
✓	i.	<p>What is “Software Engineering”?</p> <p>Use of systematic, engineering approach in all stages of software development and project management to develop high quality and economical software using appropriate software tools.</p>	(04 Marks)
✓	ii.	<p>Give three reasons why software need to be maintained.</p> <ul style="list-style-type: none">• Errors in the system.• Changes in the user requirements• Availability of new technology• Changes in the enterprise or Govt. policy. <p>Any three reasons. 01 marks x 3= 03 marks</p>	(03 Marks)
✓	iii.	<p>“Software engineering is different compared to other engineering disciplines.” Do you agree with this statement? Explain your view. (Hint: use at least three reasons)</p> <p>Yes ----- 01 marks</p> <ul style="list-style-type: none">• It is difficult for a customer to specify requirements completely.• It is difficult for the developer to understand fully the customer needs.• Software requirements change regularly.• Software is primarily intangible; much of the process of creating software is also intangible, involving experience, thought and imagination.• It is difficult to test software exhaustively. <p>Any 03 reason: 02 x 3= 06 marks</p>	(07 Marks)

✓	iv.	<p>There are two classifications for software quality attributes as Bohem's Classification and McCall's Classification. Name two attributes of each classification and give an example for each attribute.</p> <p>Categories of Bohem's Classification</p> <p>Current usefulness 01 mark</p> <p>Exampleany one example of below 01 mark</p> <ul style="list-style-type: none"> • Efficiency • Reliability • Usability • Correctness • User friendliness • Robustness <p>Potential Usefulness 01 mark</p> <p>Exampleany one example of below 01 mark</p> <ul style="list-style-type: none"> • Maintainability • Modularity • Reusability • Portability <p>Categories of McCall's Classification;</p> <p>Any two categories given bellow: 01 Mark x2 =02 Marks</p> <p>Any one example from each categories given bellow:..... 01 Mark x2 =02 Marks</p> <p>Category: Product operation</p> <p>Example</p> <ul style="list-style-type: none"> • Efficiency • Correctness • User friendliness • Usability • Reliability • Robustness <p>Category: Product revision</p> <p>Example</p> <ul style="list-style-type: none"> • Maintainability • Flexibility 	(08 Marks)
---	-----	---	------------

		<ul style="list-style-type: none">• Testability <p>Category: Product transition</p> <p>Example</p> <ul style="list-style-type: none">• Interoperability• Reusability• Portability									
✓	v.	Match the most suitable expecting goal with following software system? <table><thead><tr><th><u>Software System</u></th><th><u>Goal</u></th></tr></thead><tbody><tr><td>a) Banking system</td><td>P). Responsive</td></tr><tr><td>b) Telephone switching system</td><td>Q). Secure</td></tr><tr><td>c) Interactive game</td><td>R). Reliable</td></tr></tbody></table> <p>a). ----- > Q) 01 mark b). ----- > R) 01 mark c). ----- > P) 01 mark</p>	<u>Software System</u>	<u>Goal</u>	a) Banking system	P). Responsive	b) Telephone switching system	Q). Secure	c) Interactive game	R). Reliable	(03 Marks)
<u>Software System</u>	<u>Goal</u>										
a) Banking system	P). Responsive										
b) Telephone switching system	Q). Secure										
c) Interactive game	R). Reliable										
2.											
✓	i.	The spiral model is divided into four main task regions. Name three of them. Determine goals, alternatives, constraints <ul style="list-style-type: none">• Evaluate alternatives and risks• Develop and test• Plan <p>Any 3 ➔ 01 x 3 =03 marks</p>	(03 Marks)								
✓	ii.	List down three principles of agile methods. <ul style="list-style-type: none">• Customer involvement,• Incremental delivery• People not process• Embrace change• Maintain simplicity <p>01 mark for one principle x3= 03 marks</p>	(03 marks)								
✓	iii.	Your development team has received three projects given as follow. Which	(03 Marks)								

		<p>project can use waterfall model?</p> <p>a) Aircraft system b) Word processing package c) Bridge designing system</p> <p>Aircraft system: Waterfall Model Suitable. 01 mark</p> <p>Word processing package: Not suitable01 mark</p> <p>Bridge designing system: Waterfall Model Suitable...01 mark</p>	
✓	iv.	<p>Evolutionary prototyping and waterfall model are two software process models. For safety-critical projects which model is more suitable? Justify your answer.</p> <p>Waterfall is more suitable.2 marks</p> <p>Reason:</p> <p>Waterfall approach is suitable for developing safety critical systems which should be precisely specified before the design process starts. 1 mark</p> <p>But in evolutionary prototype starts with poorly understood requirements. Once the requirements are cleared, the system will be modified and added the new features to the next prototype during development. 1 mark</p>	(04 Marks)
✓	v.	<p>Briefly explain the differences and similarities between evolutionary prototyping and incremental approaches in systems development.</p> <p>Both approaches i.e. evolutionary prototyping and incremental development are iterative.(02 marks)</p> <p>Evolutionary prototyping involves the creation of a number of versions of the prototype system, which will be transformed into the final working system.(02 marks)</p> <p>Incremental development is an approach where the software is delivered and deployed in increments. Each increment provides a sub-set of the system functionality. ... (02 marks)</p>	(06 Marks)
✓	vi.	<p>Discuss the characteristics of software development projects which prototyping would be suitable.</p>	(06 Marks)

		<p>Systems that will involve numerous interactions with the user, in particular those where the user can determine the sequence of events e.g. websites. (3 marks)</p> <p>Systems where the requirements may be vague or uncertain, or where the requirements may change during development..... (3 marks)</p>	
3.			
✓	i.	<p>Name four processes in Requirements Engineering.</p> <p>Feasibility Study</p> <p>Requirement elicitation and analysis</p> <p>Requirements specification</p> <p>Requirements validation</p>	(04 Marks)
✓	ii.	<p>Briefly explain two types of Requirements.</p> <p>Functional Requirements; these are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.</p> <p>Non-functional Requirements; These are constraints on the services or functions offered by the system.</p> <p>Domain Requirements: These are requirements that come from the application domain of the system and that reflect characteristics of the domain.</p> <p>User requirements; Statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for customers.</p> <p>System requirements: A structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor</p> <p>For any two of above. 01 mark for name it and 01 mark for explain it.</p>	(04 Marks)
✓	iii.	<p>Briefly explain two requirements validation techniques.</p> <p><u>Requirements Reviews</u> - The requirements are analyzed systematically by</p>	(06 marks)

		<p>a team of reviewers.</p> <p><u>Prototyping</u> - In this approach to validation, an executable model of the system is demonstrated to end-users and customers. They can check whether the requirements satisfy their needs</p> <p><u>Test-case generation</u> - Requirements should ideally be testable. If a test is difficult or impossible to design, this usually means that the requirements will be difficult to implement and should be reconsider.</p> <p><u>Automated consistency analysis</u>: If the requirements are expressed as a system model in a structured or formal notation than CASE tools may be used to check the consistency of the model</p> <p>For any two of above. 01 mark for name it and 02 mark for explain it.</p>	
✓	iv.	<p>Following steps explain the “customer is paying by cash” scenario in Point-of-Sale system.</p> <ul style="list-style-type: none"> ▪ Cashier enters the customer paid amount to system through keyboard. ▪ System presents the balance due and release the cash drawer. ▪ Cashier deposits the cash paid by customer and returns balance in cash to customer. ▪ System records the cash payment. <p>Complete the following part of form which used in form based approach in requirements specification for the above scenario?</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p>Function:(a).....</p> <p>Description: In the POS (Point-Of-Sale), after buy things customer need to pay the money (amount) tendered</p> <p>Inputs:.....(b).....</p> <p>Source:(c).....</p> <p>Outputs: (d).....</p> <p>Destination:(f).....</p> </div>	(05 Marks)

Function: (a).....Customer paying by cash.....

Description: In the POS (Point-Of-Sale), after buy things customer need to pay the money (amount) tendered

Inputs: (b).customer paid amount / cash received from customer.....

✓ v.	<p>Imagine you have to develop an on-line Patient Health Record Management System. That system will be used to maintain information about patients and receiving treatments for their health problems. List down three functional requirements for the above system.</p> <p><u>Functional requirements:</u></p> <p>Any two services from system can accept as answer (02 mark x3 =06 marks)</p> <p>Sample answers:</p> <ul style="list-style-type: none">▪ Patient should be able to register with the system on line▪ Patient should be able to search his/ her health history details by NIC▪ The doctor should be able to update the patient health records	(06 Marks)	
4.			
✓ i.	<p>Name four Software Design activities.</p> <ul style="list-style-type: none">• Identification of the sub-systems• Identification of the software components• Identification of the software architecture• Data design• Interface design• Algorithm design• Data structure design• Design specification	(04 marks)	

		Any 04 reason. 01 mark for one reason 01 x 4= 04 marks	
✓	ii.	<p>Briefly explain the following Design Principles</p> <p>(a) Abstraction (b) Encapsulation (c) Loose coupling (d) Module Cohesion</p> <p>(a)<u>Abstraction</u>: This is an intellectual tool (a psychological notion) which permits one to concentrate on a problem at some level of generalization without regard to irrelevant low level details</p> <p>(b)<u>Encapsulation</u>: Protecting information from direct access by other modules and providing access to this information through well-defined interfaces is called Encapsulation.</p> <p>(c)<u>Loose coupling</u>: Loose coupling means component changes are unlikely to affect other components.</p> <p>(d)<u>Module Cohesion</u>: Interaction within a module. A measure of how well a component fits together.</p> <p>02 marks x4 = 08 marks</p>	(08 Marks)
✓	iii.	<p>Briefly explain the difference between followings:</p> <p>a). Adaptive Maintenance and Perfective Maintenance.</p> <p>b). Software Re-engineering and Reverse engineering</p> <p>a). Adaptive Maintenance (adapting the software to new environments)</p> <p style="padding-left: 40px;">Maintenance to add to or modify the system's functionality Modifying the system to satisfy new requirements Modifying the system to suit new operation environment</p> <p>Perfective Maintenance</p> <p style="padding-left: 40px;">Improving programs performance, structure, reliability etc. Making changes to avoid future problems or prepare for future changes</p> <p>02 marks for explaining each → 02x 2=04 marks</p> <p>b). re-engineering Re-structuring or re-writing part or all of an existing system without changing its functionality is called re-engineering</p>	(08 Marks)

		reverse engineering: <ul style="list-style-type: none">Analysing software to gain an understanding of its design and specificationMay be part of a re-engineering process or to re-specify a system for re-implementation							
✓	iv.	<p>“Software Designing is an important process in system Development Life Cycle”. Do you agree with this statement? Briefly explain by giving two reasons.</p> <p>Yes -----01 mark</p> <ul style="list-style-type: none">A good design is the key for a successful software systemA good design allows easy maintenance of a systemA good design allows to achieve non-functional requirements such as reliability, performance, reusability, portability.A good design facilitates the development and management processes of a software project. <p>Any two reasons. 02 x2 =04 marks</p>	(05 Marks)						
	5.								
✓	i.	<p>Testing procedures should be established at the start of any software project. All testing carried out should be based on a test plan. Name three factors which should be included in the test plan.</p> <ul style="list-style-type: none">The pre-requisites for the tests.The steps required to carry out the testsThe expected results of the test.	(03 Marks)						
✓	ii.	<p>Write the author and a technique which is used for following test phases.</p> <p>a). Unit Test</p> <p>b). Integration Test</p> <p>c). System Test</p> <table><tr><td>Test Phase</td><td>Author</td><td>Technique</td></tr><tr><td></td><td></td><td></td></tr></table>	Test Phase	Author	Technique				(03 marks)
Test Phase	Author	Technique							

		<table> <tr> <td>Unit Test</td><td>Designer 0.5 mark</td><td>White Box, Black box, Static Any one :0.5 mark</td><td></td></tr> <tr> <td>Integration Test</td><td>Author of specification 0.5 mark</td><td>Black box, white box, Top-down, bottom-up Any one: 0.5 mark</td><td></td></tr> <tr> <td>System Test</td><td>Analyst 0.5 mark</td><td>Black box, stress testing ,performance testing Any one: 0.5 mark</td><td></td></tr> </table>	Unit Test	Designer 0.5 mark	White Box, Black box, Static Any one :0.5 mark		Integration Test	Author of specification 0.5 mark	Black box, white box, Top-down, bottom-up Any one: 0.5 mark		System Test	Analyst 0.5 mark	Black box, stress testing ,performance testing Any one: 0.5 mark		
Unit Test	Designer 0.5 mark	White Box, Black box, Static Any one :0.5 mark													
Integration Test	Author of specification 0.5 mark	Black box, white box, Top-down, bottom-up Any one: 0.5 mark													
System Test	Analyst 0.5 mark	Black box, stress testing ,performance testing Any one: 0.5 mark													
✓	iii.	<p>Name three activities in Software Project Management.</p> <p>Any three 01mark x 3= 0-3 marks</p> <ul style="list-style-type: none"> • Proposal writing • Project planning and scheduling • Project costing • Project monitoring and reviews • Personnel selection and evaluation • Report writing and presentation 	(03 Marks)												
✓	iv.	<p>Name milestones of following processes</p> <p>a). feasibility study</p> <p>b). prototype development</p> <p>c). Requirement specification</p> <p>a). feasibility study Feasibility report</p> <p>b). prototype developmentEvaluation report</p> <p>c). Requirement specificationsystem requirements.</p>	(03 marks)												
✓	v.	<p>List down three basic techniques for Component (Version) identification?</p> <p>Version numbering</p> <p>Attribute-based identification</p>	(03 Marks)												

		Change-oriented identification																			
✓	vi.	<p>Briefly explain drawbacks in manual version control</p> <ul style="list-style-type: none">• Difficult to separate permanent updates from experimental changes• Very complex if you are dealing with multiple documents such as programs, word documents manuals• When multiple people are working together it is difficult to find the difference between the versions and also difficult to merge them to a single document	(03 marks)																		
✓	vii.	<p>Draw the Activity on node, network diagram and find the critical path for following data</p> <table><tr><th>Tasks</th><th>Predecessors Task (Dependencies)</th><th>Time/Duration (weeks)</th></tr><tr><td>A</td><td>-</td><td>5</td></tr><tr><td>B</td><td>A</td><td>3</td></tr><tr><td>C</td><td>A</td><td>4</td></tr><tr><td>D</td><td>B, C</td><td>2</td></tr><tr><td>E</td><td>D</td><td>3</td></tr></table> <pre>graph LR; Start[Start] -- 5 weeks --> A[A]; A -- 3 weeks --> B[B]; A -- 4 weeks --> C[C]; B --> D[D]; C --> D; D -- 2 weeks --> E[E]; E -- 3 weeks --> End[End]</pre> <p>03 marks. 0.5 mark for one node and 0.5 mark for start and end node 0.5 x 5 =2.5 marks 2.5 marks + 0.5 = 03 marks</p> <p>There are two paths</p> <p>A – B – D –E = 5 + 3 + 2 + 3 = 13 weeks 01 mark</p> <p>A – C – D –E = 5 + 4 + 2 + 3 = 14 weeks 01 mark</p> <p>Highest duration is 14 weeks.</p> <p>Therefore critical path is : A-C-D-E 02 marks</p>	Tasks	Predecessors Task (Dependencies)	Time/Duration (weeks)	A	-	5	B	A	3	C	A	4	D	B, C	2	E	D	3	(07 Marks)
Tasks	Predecessors Task (Dependencies)	Time/Duration (weeks)																			
A	-	5																			
B	A	3																			
C	A	4																			
D	B, C	2																			
E	D	3																			