SSN COLLEGE OF ENGINEERING, KALAVAKKAM – 603 110
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
B.E. Computer Science and Engineering
CS6403 Software Engineering

Date: 02.02.2018, 8.00-9.30 AM **UNIT TEST – 1** **Max. Marks: 50**
Academic Year: 2017-2018 EVEN **Batch: 2016-2020**
Semester: 4 **Faculty: Dr. R. Kanchana and Dr. A. Chamundeswari**

Qn.No	Part – A (5 * 2 = 10)	Marks	(KL,COn)
1	Define software crisis. The situation where the quality of software generally was unacceptably low and that deadlines and budgets were not being met. Only 35 percent of the projects were successfully completed, whereas 19 percent were canceled before completion or were never implemented. The remaining 46 percent of the projects were completed and installed on the client's computer. However, those projects were over budget, late, or had fewer features and functionality than initially specified. In other words, during 2006, just over one in three software development projects was successful; almost half the projects displayed one or more symptoms of the software crisis.	2	K1,CO1
2	Analyse the impact of discovering errors in the later phases of SDLC. Refer page 13 Fig 1.5 and corresponding narrative of Schcha	2	K2,CO1
3	Distinguish between CPM and PERT techniques for project scheduling.	2	K2,CO1

BASIS FOR COMPARISON	PERT	CPM
Meaning	PERT is a project management technique, used to manage uncertain activities of a project.	CPM is a statistical technique of project management that manages well defined activities of a project.
What is it?	A technique of planning and control of time.	A method to control cost and time.
Orientation	Event-oriented	Activity-oriented
Evolution	Evolved as Research & Development project	Evolved as Construction project
Model	Probabilistic Model	Deterministic Model
Focuses on	Time	Time-cost trade-off
Estimates	Three time estimates	One time estimate
Appropriate for	High precision time estimate	Reasonable time estimate

Management of	Unpredictable Activities	Predictable activities
Nature of jobs	Non-repetitive nature	Repetitive nature
Critical and Non-critical activities	No differentiation	Differentiated
Suitable for	Research and Development Project	Non-research projects like civil construction, ship building etc.
Crashing concept	Not Applicable	Applicable

4 Mention the need for task network. 2 K2,CO5

A task network, also called an activity network, is a graphic representation of the task flow for a project. It is sometimes used as the mechanism through which task sequence and dependencies are input to an automated project scheduling tool. In its simplest form (used when creating a macroscopic schedule), the task network depicts major software engineering actions.

5 Describe the differences between "*known risks*" and "*predictable risks*". 2 K2,CO5

Known risks are those that are determining through careful evaluation of the project and technology. Predictable risks are extrapolated from past experience.

Part – B Answer all questions (13+13)

6 a) What are the shortcomings of a present software development process? 3 K2,CO1

Despite many software success stories, an unacceptably large proportion of software products still are being delivered late, over budget, and with residual faults.

In a survey conducted by the Cutter Consortium [2002], the following was reported:

• An astounding 78 percent of information technology organizations have been involved in disputes that ended in litigation.

• In 67 percent of those cases, the functionality or performance of the software products as delivered did not measure up to the claims of the software developers.

• In 56 percent of those cases, the promised delivery date slipped several times.

• In 45 percent of those cases, the faults were so severe that the software product was unusable.

Many projects exhibit moving target problem

b) State the objectives of software engineering. How can the present software development process be improved by using software engineering principles? 1+2 K2,CO1

Software engineering is defined as a discipline whose aim is

the production of fault-free
software that satisfies the user's needs and is delivered on time and within budget. To achieve this goal, appropriate techniques have to be used throughout software production, including when performing analysis (specification) and design and post-delivery maintenance . Software engineering addresses all the steps of the software life cycle and incorporates aspects of many different areas of human knowledge, including economics and the social sciences.
software engineering uses the philosophies and paradigms of established engineering disciplines to solve software crisis.
It is clear that far too little software is delivered on time, within budget, fault free, and meeting its client's needs. To achieve these goals, a software engineer has to acquire a broad range of skills, both technical and managerial. These skills have to be applied not just to programming but to every step of software production, from requirements to postdelivery maintenance.
Since postdelivery maintenance is so important, a major aspect of software engineering consists of those techniques, tools, and practices that lead to a reduction in postdelivery maintenance costs.
Software professionals are human and therefore sometimes make a mistake while developing a product. As a result, there will be a fault in the software. If the mistake is made while eliciting the requirements, the resulting fault will probably also appear in the specifications, the design, and the code. Clearly, the earlier we correct a fault, the better.

c) List any FIVE life cycle models. Tabulate their advantages and disadvantages. Map each of them to a respective process model. 1+5+1 K2,CO1

Life-Cycle Model	Strengths	Weaknesses
Evolution-tree model (Section 2.2)	Closely models real-world software production Equivalent to the iterative-and-incremental model	
Iterative-and-incremental life-cycle model (Section 2.5)	Closely models real-world software production Underlies the Unified Process	
Code-and-fix life-cycle model (Section 2.9.1)	Fine for short programs that require no maintenance	Totally unsatisfactory for nontrivial programs
Waterfall life-cycle model (Section 2.9.2)	Disciplined approach Document driven	Delivered product may not meet client's needs
Rapid-prototyping life-cycle model (Section 2.9.3)	Ensures that the delivered product meets the client's needs	Not yet proven beyond all doubt
Open-source life-cycle model (Section 2.9.4)	Has worked extremely well in a small number of instances	Limited applicability Usually does not work
Agile processes (Section 2.9.5)	Work well when the client's requirements are vague	Appear to work on only small-scale projects
Synchronize-and-stabilize life-cycle model (Section 2.9.6)	Future users' needs are met Ensures that components can be successfully integrated	Has not been widely used other than at Microsoft
Spiral life-cycle model (Section 2.9.7)	Risk driven	Can be used for only large-scale, in-house products Developers have to be competent in risk analysis and risk resolution

Evolution tree – Sequence model
Iterative and increment / waterfall – Iterative model
RP, spiral – Prototyping model
Synchronize and stabilize – concurrent model

OR

7 a) Suggest the most appropriate generic software process model that might be used as a basis for managing the development of thefollowing systems. Give reasons for your answer. 8 K4,CO1

- A system to control anti-lock braking in a car
- A virtual reality system to support software maintenance
- A university accounting system that replaces an existing system
- An interactive travel planning system that helps users plan journeys with the lowest environmental impact

1. *Anti-lock braking system* This is a safety-critical system so requires a lot of up-front analysis before implementation. It certainly needs a plan-driven approach to development with the requirements carefully analysed. An iterative or waterfall model is therefore the most appropriate approach to use, perhaps with formal transformations between the different development stages.

2. Virtual reality system This is a system where the requirements will change and there will be an extensive user interface components. Incremental development with, perhaps, some UI

prototyping is the most appropriate model. An agile process may be used.

3. University accounting system This is a system whose requirements are fairly well-known and which will be used in an environment in conjunction with lots of other systems such as a research grant management system. Therefore, a reuse-based approach is likely to be appropriate for this.

4. Interactive travel planning system System with a complex user interface but which must be stable and reliable. An incremental development approach is the most appropriate as the system requirements will change as real user experience with the system is gained.

b) Assume that you are a software engineering consultant and have been called in by the vice-president for finance of a corporation that manufactures tires and sells them via its large chain of retail outlets. She wants your organization to build a product that will monitor the company's stock, starting with the purchasing of the raw materials and keeping track of the tires as they are manufactured, distributed to the individual stores, and sold to customers. 2+2+1 K4,CO1

i. Identify a suitable life cycle model for this product development.
Answer varies

ii. What criteria would you use in selecting a life-cycle model for the project?
Experience and skills of the development team; computer literacy of the client; extent to which the client seems to appreciate his or her real needs.

iii. List the risks involved in developing the software. How would you attempt to mitigate each risk?
The product may not be what the client really needs, so construct a rapid prototype. The design may not permit future development as the corporation grows or changes the way it does business, so ensure that the design is as open-ended as is reasonable. There may be cost and or time overruns, so estimate carefully. The users may not be comfortable with the product, so a rapid prototype of the user interface is needed; also, involve the purchasing clerks, factory supervisors, sales clerks, and so on, in the development loop. A competitor may produce off-the-shelf software before the product has been delivered — there is no ethical way to resolve this risk. A critical member of the development team may leave, so keep management of the development organization abreast of major decisions being made, thereby making it easier to integrate a replacement into the team. The development team may not be properly managed, so ensure that managers are competent and well-trained.

8 a) List the differences between *process metrics and project metrics*. 3 K3,CO5

A process metric is used to assess the activities that are used to engineer and build computer software (with the intent of improving those activities on subsequent projects). A project metric is used to assess the status of a software project

b) Suppose a 40 KLOC software product can be purchased for 5lakh. Assuming that in-house programmers' cost is 10000 per PM including overhead, would it be more cost-effective to buy the product or to build it? What additional factors should be considered in making this make/buy decision? 4+1 K3,CO5

Answer varies

c) It seems odd that cost and schedule estimates are developed during software project planning, before detailed software requirements analysis or design has been conducted. Why do you think this is done? Are there circumstances when it should not be done? 3+2 K4,CO5

Costs and schedule are estimated early because such information is demanded (by upper management) as early as possible. If a project is extremely complex with high technological risk and a fixed price proposal is to be submitted, costing of the project should (if possible) be delayed until after requirements analysis. Note: the cost of requirements analysis alone can be estimated early.

OR

9 a) Write short notes on empirical estimation models for software projects. 8 K2,CO5

Refer Pressman chapter 26.7 COCOMO, COCOMO II

b) Write short notes on *Earned Value Analysis (EVA)* that monitors and assesses software project schedules. 5 K2,CO5

Refer Pressman chapter 27.6

Part – C (14)

10 a) A target product has 8 simple inputs, 3 average inputs, and 11 complex inputs. There are 57 average outputs, 9 simple inquiries, 13 average master files, and 18 complex interfaces. 4 K3,CO5

i. Determine the unadjusted function points (*UFP*).
724

ii. If the value adjustment factor (degree of influence) is 47, determine the number of function points.
810.88

iii. Why do you think that, despite its drawbacks, lines of code (LOC or KDSI) is so widely used as a metric of product size?

LOC is an "artifact" of all software development projects that can be easily counted, that many existing software estimation models use LOC or KLOC as a key input, and that a large body of literature

i. and data predicated on LOC already exists

Component	Level of Complexity		
	Simple	Average	Complex
External Inputs	3	4	6
External outputs	4	5	7
External Inquiries	3	4	6
Internal logical files	7	10	15
External interface files	5	7	10

Table 1: Weighing factor for Information domain values

b) Use the project data given below to prepare a time line chart and calculate the total time taken to complete the project. 2+3 K3,CO5

Task	Predecessor	Duration(in days)
A	-	2
B	A	2
C	B	1
D	A	1
E	B,D	4
F	C,E	2

Time taken - 10 days

c) Assume that you are the project manager for a major software company and you have been asked to lead a team that is developing "next generation" word-processing software. Create a **risk table** for the project. — 5 — K3,CO5

Business risks here are probably the most relevant, although students will likely select technical and project risks. Be sure to indicate that business risks (i.e., large market share of existing WPs with significant penetration problems); industry standard WPs (e.g., MSWord) will cause reticence to buy, etc. may be significantly more important than technical or project risks.

*************B E S T O F L U C K************

Prepared by	
R. Kanchana	A. Chamundeswari

Reviewed by HoD, CSE