



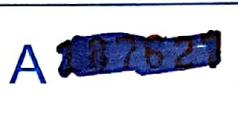
# OLD PAPERS FOR PROJECT USE ONLY

## PES INSTITUTE OF TECHNOLOGY, BANGALORE

(AUTONOMOUS INSTITUTE UNDER VTU, BELGAUM & UGC, NEW DELHI)



SERIAL NUMBER



BE / MBA / MCA / M.Tech. DEGREE EXAMINATION

TEST ANSWER BOOKLET

**USN**

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

**BRANCH**

CSE
-----

**SEMESTER**

6/12
------

**DATE OF EXAMINATION**

1	9	/	1	0	2	/	0	0
---	---	---	---	---	---	---	---	---

**SUBJECT TITLE**

Software Engg.
----------------

**SUBJECT CODE**

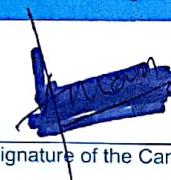
13CS354
---------

I abide by the rules and regulations of Institute.

Signature of the Room Superintendent



Signature of the Candidate



For Evaluator's use only

Question Number	Sub - Questions				Total Marks
	a	b	c	d	
1	3	1	3		07
2	5	2	2		09
3	6	3			09
4	0	1	0		01
5	6	3			09
				Grand Total	35
				Marks in words	Three five
Evaluator's Signature with Date					

Test -1

In the boxes provided for entry of marks, small check boxes are provided against each sub-division. After answering the sub-division, cross the corresponding checkbox to indicate that you have answered the same, not crossing the checkbox may lead to your answers not being valued. Please take care.

1. a)

1. False

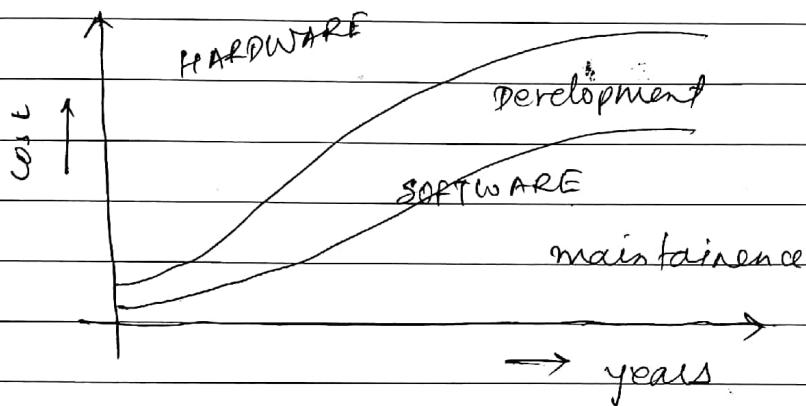
2.

3.

4.

2.

False



The above clearly depicts the current trend comprising the cost of the project bifurcated into HARDWARE & SOFTWARE.

3)

TRUE.



4.

False.

Coz software engineering is all about starting with a skeletal framework (need not have to be 100% accurate) and from there on tweaking the software bit by bit as executing revisions on it or backtracking if need be.

So, software doesn't have to be fully correct from its inception.

1-b)

In the process of elicitation i.e. acquiring requirements specification, we would

need to learn which

→ requirements to finish first off &

→ which requirements to drop

coz maybe its redundant or  
it makes the code more

complex.

IncrementalIterativeSE models

It has to do more with adding new functionalities and features.

If It has much to do with fine-tuning what already has been done or implemented.

Project proceeds in units of small modules which are finished fast.

2. Project proceeds by iterative approach where upon, in each iteration all the features are developed but to a certain extent.

4. Easier to add new functionality to the project.

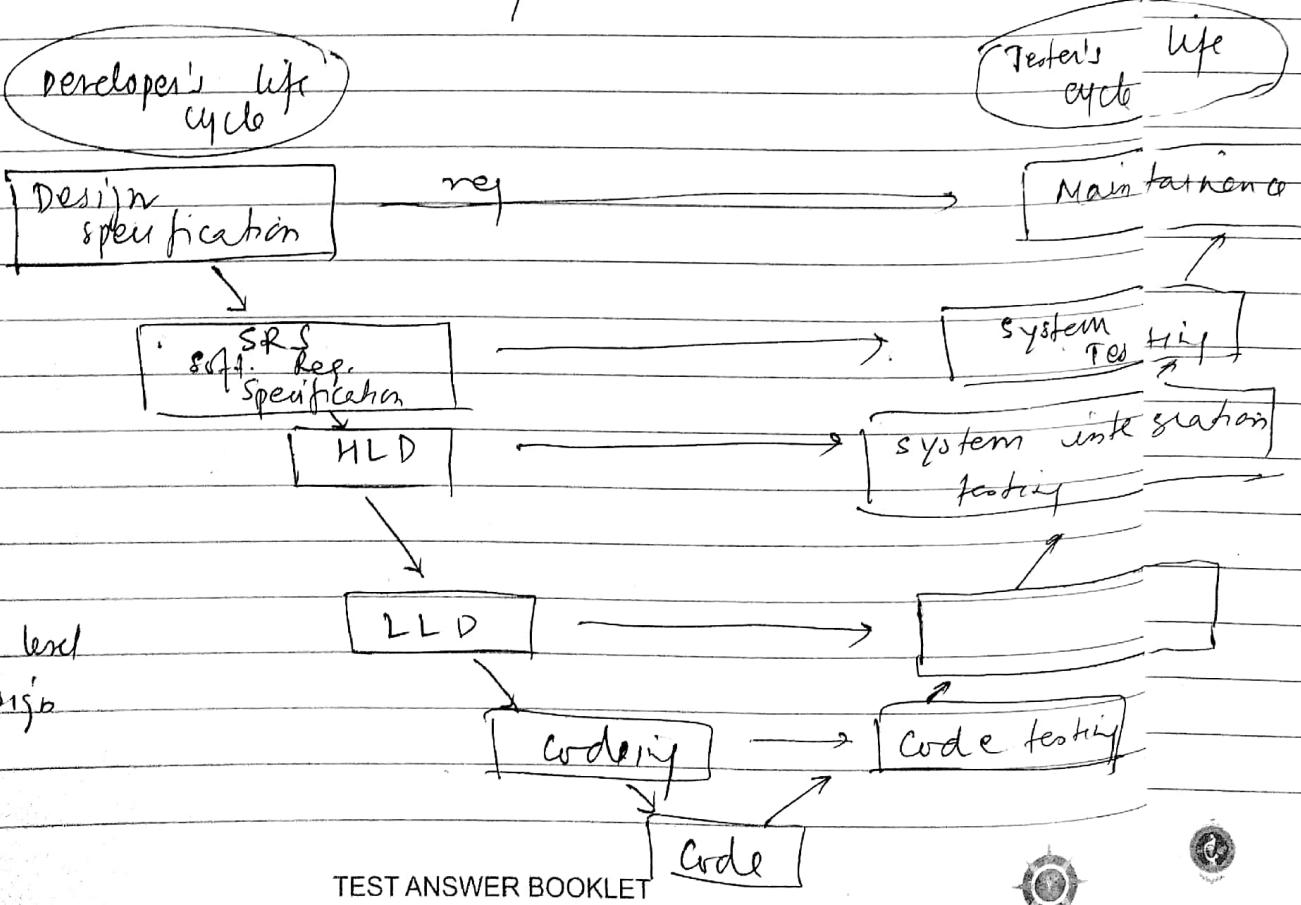


5. When adding new feature to the project, old code might have to be re-written.



5. esp: Drawing a picture part by part where each part is rendered complete in each increment

5. esp: Creating a picture as a whole by layering upon the previously done work.



→ Main difference b/w V-model and waterfall-model in the V-model being that design is done way before the actual implementation.

So, this generally results in more accurate code.

→ V-model

\* The figure a to the left depicts the phases followed in V-model software development life cycle.

It starts with design specification evaluation along many stages which is then followed by code. (physical)

As the coding progresses, the other phases also keep progressing such as System integration etc.

So,

phases aren't totally dependent before them.

(P.7.0.6)

→ We would use v-model in the case when requirement may change afterwards or where multiple prototyping is required for fine-tuning the software. (d+c)

2.b)

### Discontinuance

1. This stage represents the scenario when the product is no more supported by the corporation or firm. i.e. if it is there in the market but no more updates or revision would be done on it.

### Obsolescence

1. This phase means that the product has become obsolete i.e. rendered inefficient in the present context, maybe by the arrival of a new product or by the advent of a new technology.

2. The company has stopped making

2. The product has lost its

The product

market & has become legacy now

Throw-away

In this kind of prototyping, if the prototype developed, does not work appropriately (the way the customer wants), it is INSTANTLY thrown away (discarded).

A new prototype is developed from scratch if the old one fails.

The philosophy or ideology which was followed to make the prototype is also trashed

evolutionary

1. If the prototype does not seem to the standard or turns out to be lacking, revisions are done onto it. i.e. the same prototype is patched or tweaked up.

2. The same old prototype is fixed and patched.

3. ~~The~~ The principle (ideology) remains the same or at least greedily is assumed to

3. a)

1)

self governing  
small team → The motivation  
behind this approach is  
that small number of people  
are easier to manage.

Making many small teams  
which due to less no. of  
people in them are self  
regulating, make the development  
faster.

Also in small teams,  
people can review each other  
efficiently.

e.g: Teams can be pairs as at  
most 3.

2.

Less Documentation → this is

based on the idea that  
more prototyping or implementation  
is preferred over extensive  
documentation.

Since less number of  
people involved in the project  
escape through meagre documentation.

e.g: Code can be documented by the  
person writing it.

TEST ANSWER BOOKLET

3. Small meeting → Instead of planning elaborate details and features at the start, agile architecture proposes quick & regular meet-ups among groups so as to be able to evaluate & integrate whatever has been covered that day. Decisions are made seeing only the current circumstance, hence, this approach is pretty flexible.

Q: Meet-ups can happen daily or even twice a day, with module integration happening towards the end of the day.

3.b)

1)

SPRINT → It is a phase that is to be covered preferably over a few weeks.

These phases are also called sprints in the sense that they profess the idea of rapid development followed with quick customer review.

These sprints are normally ended with a meet-up where customer deems if the new feature added during the sprints is acceptable.

2.

SCRUM → It is dividing a big team into small self-sustaining teams with more attention on regular efficient development among the groups.

This also removes redundancies among all the



groups & makes sure each member is put to his total use.

Backlog:- mainly divided into  
 (Sprint Backlog) & (Project Backlog).

It is the amount work left undone after the each sprint session. i.e all not done falls under this

Scrum master → A member among the whole team is made the scrum master with holding up meetings, job of laying out daily plan & interfacing project management & the team.

He also responsible for integration

~~of~~ when specifying

- 4.b)
- 1) Elicitation of information from customer, stake holder & user.
  - 2) Sitting with the user or customer and carefully going out ~~and~~ each end feature.
  - 3) Prioritizing the requirement gathered with immediate participation of the user.
  - 4) Also be clear about what THE USER DOES NOT WANT and removing the redundancies in the requirement specification.
- 4.c)

5.a)

1)

check-point → A state reached where upon certain (modules) tasks have been finished and the project has progressed.

It is the state of achievement of some specific major requirements (which cover big parts of the product).

2)

Budget → The whole capital which can be spent on the development as well as the main features of the product.

3)

Work-Break-Down → Intuitively, it describes the Work-break-down of the project, summarizing what is to be done & by whom.

This makes the development process more concrete & less abstract.



Empirical → This is a formula based estimation technique which has been found to be true for most cases (hence empirical)

a <sub>a</sub>	b <sub>b</sub>	c <sub>c</sub>	d <sub>d</sub>
1.09	0.05	2.5	3.2
1.08	0.66	2.5	2.9

\* estimation done in kilo lines of code

formula →  $(kLOC)^{a_a}$

Experience-Based → This is estimating the project development time by considering the past experience of the Project Manager as the Scrum Master.

The Manager puts forth an estimation which can be changed by small proportions later.

