CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

IMPROVEMENT TEST

Sub:	SOFTWARE ENGINEERING						Code:	15CS42
Date:	25 / 05 / 2017	Duration:	90 mins	Max Marks:	50	Sem: 4	Branch:	CSE

Answer **FIVE** FULL questions selecting AT LEAST ONE question **from each part**

		Marks	OBE CO	OBE RBT
	PART A			
1 (a)	Explain software review process with a neat diagram.	[10]	CO5	L2
	OR			
2 (a)	List and explain the Inspection Checklist.	[10]	CO5	L2
	PART B			
3 (a)	Explain examples of product metrics (Static and CK suite).	[5+5]	CO5	L2
	OR			
4 (a)	What are software standards? Define 9 core processes of ISO 9001 standard.	[1+9]	CO5	L2
	PART C			
5 (a)	With a neat diagram, explain the process of prototype development.	[10]	CO5	L2
(b)	Why Prototype should be discarded after use?	[2]	CO5	L3
	OR			
6 (a)	Explain principles of Agile development.	[12]	CO5	L2

CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

IMPROVEMENT TEST

Sub:	SOFTWARE ENGINEERING						Code:	15CS42
Date:	25 / 05 / 2017	Duration:	90 mins	Max Marks:	50	Sem: 4	Branch:	CSE

Answer **FIVE** FULL questions selecting AT LEAST ONE question **from each part**

		Marks	OBE CO	OBE RBT
	PART A			
1 (a)	Explain software review process with a neat diagram.	[10]	CO5	L2
	OR			
2 (a)	List and explain the Inspection Checklist.	[10]	CO5	L2
	PART B			
3 (a)	Explain examples of product metrics (Static and CK suite).	[5+5]	CO5	L2
	OR			
4 (a)	What are software standards? Define 9 core processes of ISO 9001 standard.	[1+9]	CO5	L2
	PART C			
5 (a)	With a neat diagram, explain the process of prototype development.	[10]	CO5	L2
(b)	Why Prototype should be discarded after use?	[2]	CO5	L3
	OR			
6 (a)	Explain principles of Agile development.	[12]	CO5	L2

	PART D			
7 (a)	List and explain practices of extreme programming.	[10]	CO5	L2
	OR			
8 (a)	Explain the SCRUM process in detail.	[10]	CO5	L2
	PART E			
9 (a)	Differentiate between:			
	i) Quality Assurance and Quality Control	[2]	CO5	L1
	ii) Software Measurement and Software Metrics	[2]		
(b)	Define:			
	i) Scaling up and Scaling out	[2]	CO5	L1
	ii) Process and Product standards	[2]		
	OR			
10 (a)	Explain pair programming and its advantages.	[5]	CO5	L2
(b)	Why software process cannot be standardized?	[3]	CO5	L3

	PART D			
7 (a)	List and explain practices of extreme programming.	[10]	CO5	L2
	OR			
8 (a)	Explain the SCRUM process in detail.	[10]	CO5	L2
	PART E			
9 (a)	Differentiate between:			
	iii) Quality Assurance and Quality Control	[2]	CO5	L1
	iv) Software Measurement and Software Metrics	[2]		
(b)	Define:			
	iii) Scaling up and Scaling out	[2]	CO5	L1
	iv) Process and Product standards	[2]		
	OR			
10 (a)	Explain pair programming and its advantages	[5]	CO5	L2
(b)	Why software process cannot be standardized?	[3]	CO5	L3

CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | |

IMPROVEMENT TEST

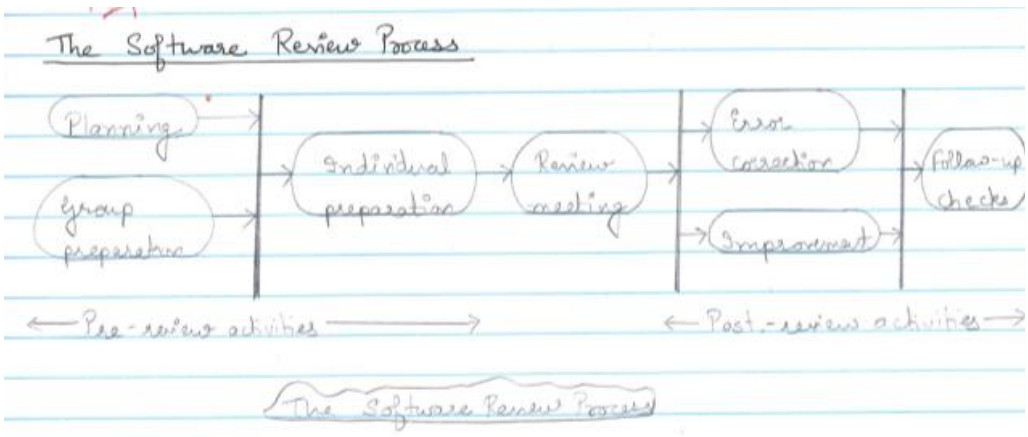
Sub:	SOFTWARE ENGINEERING							Code:	15CS42
Date:	25 / 05 / 2017	Duration:	90 mins	Max Marks:	50	Sem:	4	Branch:	CSE

Answer **FIVE** FULL questions selecting AT LEAST ONE question **from each part**

	Marks	OBE CO	OBE RBT

PART A

1 (a) Explain software review process with a neat diagram. (Diagram- 7M, Explanation- 3M) [10] CO5 L2



Review process has three phases :

(i) Pre-review activities – It contains review planning and preparation. Review planning involves setting up a review team, arranging a time and place for review, and distributing the documents to be reviewed. In review preparation, individual team members read and understand the software or documents and relevant standards to find errors, omissions, and departures from standards. If meeting can't be attended by someone, they can write comments.

(ii) The review meeting – Review should be relatively short – two hours at most. An author of document or program being reviewed should 'walk through' the document. One team member should chair the review and another should formally record all review decisions and actions to be taken. All written comments should be considered. The review chair should sign a record of comments and actions during the review.

(iii) Post review activities – Issues & problems raised during the review must be addressed (fixing bugs, refactoring, rewriting).

Sometimes, a further review will be required to check that changes made cover all the previous review comments

The review process in agile software development is usually informal. In Scrum, there is a review meeting after each iteration of the software has been completed (a sprint review) where quality issues and problems may be discussed. In extreme programming, pair programming ensures that code is constantly being examined and reviewed by another team member.

OR

2 (a) List and explain the Inspection Checklist. [10] CO5 L2

Inspection checklist

Fault Class	Inspection check
Data faults	• Are all program variables initialized before their values are used? • Have all constants been named? • Should the upper bound of arrays be equal to the size of the array or size-1? • If character strings are used, is a delimiter explicitly assigned? • Is there any possibility of buffer overflow?
Control faults	• For each conditional statement, is the condition correct? • Is each loop certain to terminate? • Are compound statements correctly bracketed? • In case statements, are all possible cases accounted for? • If a break is required after each case in case statements has it been included?
Input/output faults	• Are all input variables used? • Are all output variables assigned a value before they are output? • Can unexpected inputs cause corruption?
Interface faults	• Do all function and method calls have the correct number of parameters? • Do formal and actual parameter type match? • Are parameters in the right order? • If components access shared memory, do they have the same model of the shared memory structure?

Storage management faults	• If a linked structure is modified, have all links been correctly reassigned? • If dynamic storage is used, has space been allocated correctly? • Is space explicitly deallocated after it is no longer required?
Exception management faults	• Have all possible error conditions been taken into account?

PART B

3 (a) Explain examples of product metrics (Static and CK suite). [5+5] CO5 L2
(Any 10- 10M)

(i) Fan-in / Fan-out
Fan-in is measure of no. of functions that call another function.
Fan-out is no. of functions that are called ~~out~~ by another function.

(ii) Length of Code
It is measure of size of a program

(iii) Cyclomatic complexity
It is measure of control complexity of program

(iv) Length of identifiers
It is measure of average length of identifiers (names for variables, classes, methods etc.)

(v) Depth of conditional nesting
This is a measure of the depth of nesting of if-statements in a program

(vi) Fog index
This is a measure of the average length of words and sentences in documents.

Chidamber and Kemerer's (CK) Suite of Object-oriented metrics :-

(i) WMC (weighted methods per class)
No. of methods in each class, weighted by complexity of each method.

(ii) DIT (Depth of Inheritance tree)
No. of discrete levels in the inheritance tree where subclasses inherit attributes and methods from superclass

(iii) NOC (Number of children)
No. of immediate subclasses in a class.

(iv) CBO (Coupling between object classes)
Classes are coupled when methods in one class use methods or instance variables defined in a different class. CBO is a measure of how much coupling exists.
A high value of CBO means classes are highly dependent

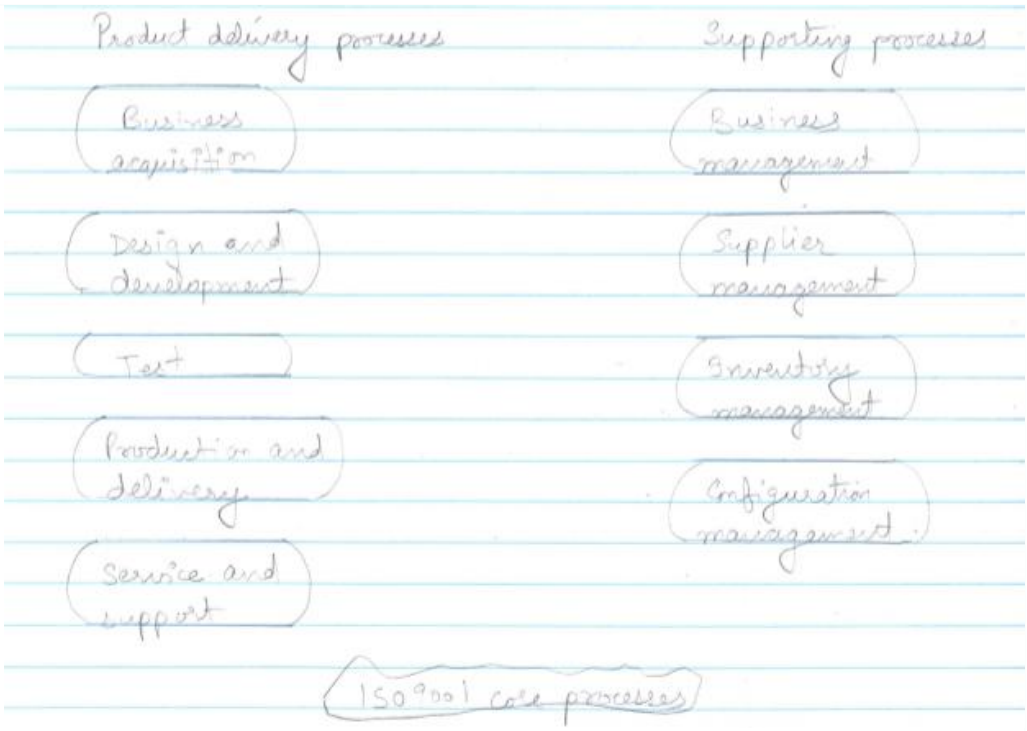
(v) RFC (Response for a Class)
No. of ~~messages~~ methods that could potentially be executed in response to a message received by an object of that Class

(vi) LCOM (Lack of cohesion in methods)
It is the difference between the number of method pairs without shared attributes and no. of method pairs with shared attributes.

OR

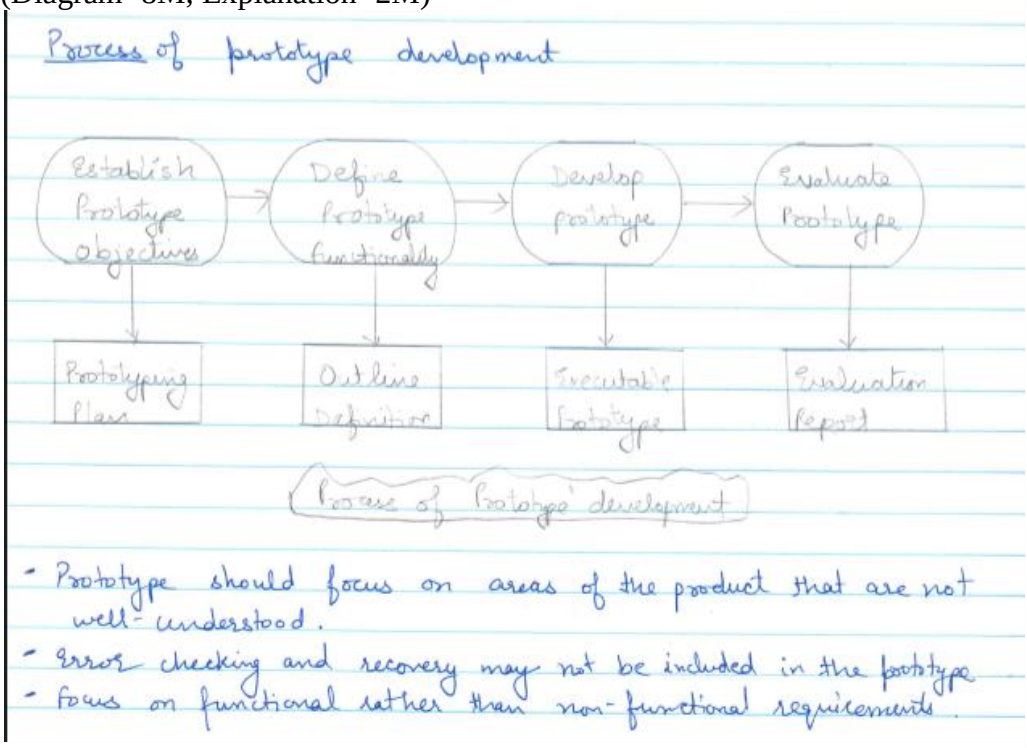
4 (a)	What are software standards? Define 9 core processes of ISO 9001 standard. (Software standards definition- 1M, Core processes- 9x1M=9M)	[1+9]	CO5	L2

Software standards define the required attributes of a product or process. They play an important role in quality management. Standards may be international, national, organizational or project standards.



PART C

5 (a) With a neat diagram, explain the process of prototype development. [10] CO5 L2
(Diagram- 8M, Explanation- 2M)



(b) Why Prototype should be discarded after use? [2] CO5 L3

Throw-away prototypes
Prototypes should be discarded after development as they are not a good basis for a production system?
- It may be impossible to tune the system to meet non-functional requirements.
- Prototypes are normally undocumented
- Prototype structure is usually degraded through rapid change
- Prototype normally will not meet normal organisational quality standards.

OR

6 (a)	Explain principles of Agile development. (Agile principles- 12x1M=12M)	[12]	CO5	L2

① Customer involvement (work together)
Customers should be clearly involved throughout the development process. Their role is to provide and prioritize new system requirements & to evaluate the iterations of the system.
② Incremental delivery (shorter timescale- weeks to months)
The software is developed in increments with the customer specifying the requirements to be included in each increment.
③ People not process (motivated individuals)
The skills of development team should be recognised and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
④ Embrace change
Expect the system requirements to change and so design system to accommodate these changes.
⑤ Maintain simplicity
Focus on simplicity in both software being developed and in the development process. Wherever possible, actively work to ~~elimin~~ eliminate complexity from the system.

⑥ Early and continuous delivery of valuable software
⑦ Most efficient and effective way of conveying information to and within a development team is face-to-face conversation.
⑧ Working software is primary measure of progress
⑨ Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
⑩ Continuous attention to technical excellence and good design enhances agility.
⑪ The best architectures, requirements, and designs emerge from self organizing teams.
⑫ At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

PART D

7 (a)	List and explain practices of extreme programming. (Extreme programming practices- 10x1M=10M)	[10]	CO5	L2

① Incremental planning

Requirements are recorded on story cards and the stories to be included in a release are determined by the time available and their relative priority. The developers break these stories into development Task's. See figures on next pages ~~to~~ for story card & task.

② Small releases

The minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally add functionality to the first release.

③ Simple design

Design is created only for the requirements to be implemented in current release.

④ Test-first development

An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented

⑤ Refactoring

All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

⑥ Pair programming

Developers work in pairs, checking each other's work and providing the support.

⑦ Collective ownership

The pairs of developers work on all areas of the system, so that no islands of expertise develop and all developers take responsibility for all of the code. Anyone can change anything

⑧ Continuous integration

As soon as the work on a task is complete, it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.

⑨ Sustainable pace

Large amounts of overtime are ~~so~~ not considered acceptable as the net effect is often to reduce code quality and medium-term productivity

⑩ On-site customer

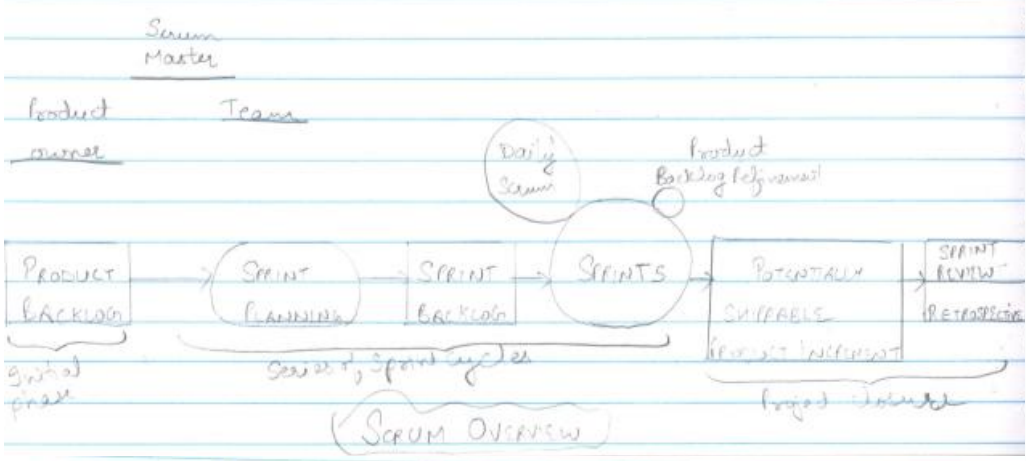
A representative of the end-user of the system (the customer) should be available full time for the use of the XP team. Customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

OR

8 (a)	Explain the SCRUM process in detail. (Diagram- 6M, Explanation- 4M)	[10]	CO5	L2

The principal responsibility of software project managers is to manage the project so that software is delivered on time and within the planned budget for the project. Agile project management requires a different approach ~~from~~ (than plan-driven approach), which is adapted to incremental development and the particular strengths of agile methods.

SCRUM



→ Scrum Roles

- Product Owner – The person responsible for managing the Product Backlog so as to maximize the value of the produ
- Team – A cross-functional group of people that is responsible for managing themselves to develop an increment of product every Sprint.
- Scrum master – He is responsible for the Scrum process, its correct implementation, and maximization of its benefits. He is a facilitator who arranges daily meetings, tracks the backlog of work to be done, records decisions, measures progress against the backlog and communicates with customers & management outside of the team.

9 (a)	→ Product Backlog The starting point of the planning is product backlog. It is a prioritized list of requirements with estimated times to turn them into completed product functionality. → Sprint planning It is a selection phase that involves all of the project team who work with the customer to select features & functionality to be developed during the sprint. → Sprint backlog It is a list of team's work for a sprint. It may be updated during sprint. → The Sprint cycle · Sprints are fixed length, normally 2-4 weeks. They correspond to the development of ~~the~~ a release of system. · Daily Scrum is a short meeting held daily by each Team during which the team members inspect their work, synchronize their work and progress. ~~one~~ They share information, describe their progress since last meeting, problems that have arisen and what is planned for the following day. This means that everyone on the team knows what is going on and, if problems arise, can re-plan short-term work to cope with them. → Sprint Review Meeting A time-boxed two hour meeting (for a two week Sprint) at the end of every sprint where Team collaborates with Product Owner and stakeholders and they inspect the output from the sprint. This usually starts with a review of completed Product Backlog items, a discussion of opportunities, constraints and risks, and a discussion of what might be the best things to do next (potentially resulting in Product Backlog changes). Only completed functionality can be demonstrated. → Sprint Retrospective Meeting A meeting facilitated by the ScrumMaster at which the complete Team discusses the just-concluded Sprint & determines what could be changed that might make the next sprint more productive.	[2] [2]	CO5	L1

PART E

Differentiate between:

i) Quality Assurance and Quality Control
(Quality Assurance- 1M, Quality Control- 1M)

Quality Assurance (QA) - It is the definition of pr and standards that should lead to high qual products and introduction of quality processes into the manufacturing process.

Quality Control - It is the application of processes specified by QA to weed out products that are no of required level of quality.

ii) Software Measurement and Software Metrics
(Software measurement- 1M, Software metrics- 1M)

Software measurement is concerned with deriving a num value for an attribute of a software product or process. This allows for objective comparison between techniques and processes. This also may highlight areas of of improvement.

A software metric is a characteristic of a software system, documentation or process that can be objectively measured. Eg - LOC, no. of reported faults etc.

(b) Define:

i) Scaling up and Scaling out [2] CO5 L1
(Scaling up- 1M, Scaling out- 1M) [2]

'Scaling up' is concerned with using agile methods for developing large software systems that cannot be devel by a small team.

'Scaling out' is concerned with how agile methods can b introduced across a large organization with many years of software development experience.

ii) Process and Product standards
(Process standards- 1M, Product standards- 1M)

Product standards define characteristics that all softwa components should exhibit e.g. a common programming Process standards define how the software process should enacted.

OR

10 (a) Explain pair programming and its advantages [5] CO5 L2
(Pair programming features- 3M, Advantages- 2M)

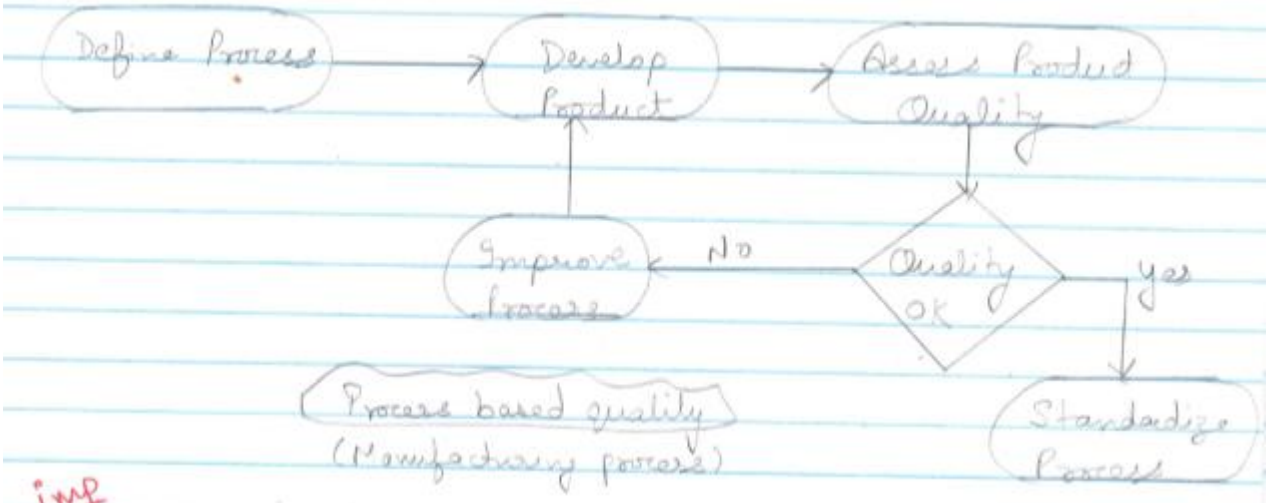
Pair Programming

- In XP, programmers work in pairs, sitting together to develop code.
- This helps develop common ownership of code and spreads knowledge across the team.
- It serves as an informal review process as each line of code is looked at by more than 1 person.
- It encourages refactoring as the whole team can benefit from this.
- Measurements suggest that development productivity with pair programming is similar to that of two people working independently.
- Programmers sit together at the same workstation to develop the software.
- Pairs are created dynamically so that all the team members work with each other during the development process.
- The sharing of knowledge that happens during pair programming is very important as it reduces the overall risks to a project when team members leave.

Advantages of pair programming:

- It supports the idea of collective ownership and responsibility for the system. Team has collective responsibility for resolving problems (not blaming).
- It acts as an informal review process because each line of code is looked at by atleast two people.
- It helps support refactoring, which is a process of software improvement.

(b)	Why software process cannot be standardized? (Diagram- 1M, Explanation- 2M)	[3]	CO5	L3



imp

However, software is not manufactured – it is designed. Therefore, relationship between process quality and product quality is more complex. S/w development is creative rather than mechanical, so influence of individual skills and experience is significant. External factors, like ~~novelty~~ novelty of an application or commercial pressure for an early product release, also affect the product quality irrespective of the process used. It is difficult to assess software quality attributes. Consequently, it is hard to tell how process characteristics influence these attributes, furthermore because of the role of design and creativity in the software process, process standardization can sometimes stifle creativity which leads to poorer rather than better quality software.