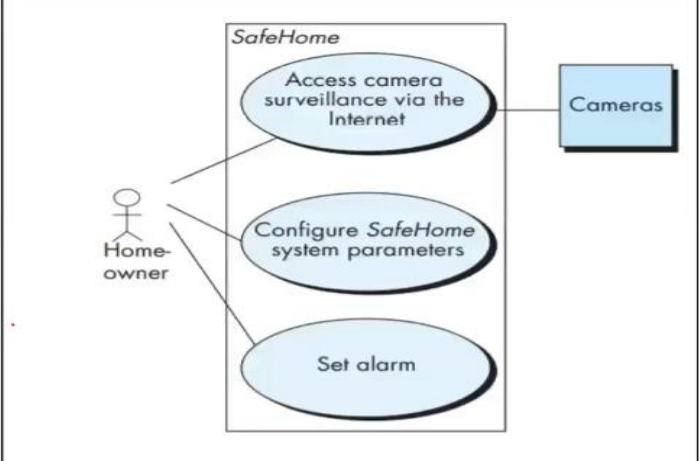


Internal Assessment Test 2 – July 2024

Sub:	SOFTWARE ENGINEERING & PROJECT MANAGEMENT	SubCode:	21CS61	Branch:	AI&DS
Date:	08/07/2024	Duration:	90 min's	MaxMarks:	50
		Sem/Sec:	VI 'A'		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT						
1	<p>Draw activity diagram and Swimlane diagram for access camera surveillance via the internet.</p> <p>Activity diagram: 5 Marks Swimlane diagram: 5 Marks</p> <div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <pre> graph TD Start(()) --> Enter[Enter password and user ID] Enter --> Valid{Valid passwords/ID} Valid -- Invalid passwords/ID --> Prompt[Prompt for reentry] Valid -- Valid passwords/ID --> Select[Select major function] Select --> SelectSurveillance[Select surveillance] Select --> Other[Other functions may also be selected] SelectSurveillance --> Input{Input tries remain} Input -- No input tries remain --> Prompt Input -- Input tries remain --> Select SelectSurveillance --> Thumbnail{Thumbnail views} Thumbnail -- Select a specific camera --> SelectIcon[Select camera icon] Thumbnail -- Select specific camera - thumbnails --> SelectIcon SelectIcon --> View[View camera output in labeled window] View --> PromptView[Prompt for another view] PromptView --> Exit{Exit this function} PromptView -- See another camera --> SelectSurveillance </pre> </div> <div style="border: 1px solid black; padding: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%;">Homeowner</th> <th style="width: 33%;">Camera</th> <th style="width: 33%;">Interface</th> </tr> </thead> <tbody> <tr> <td style="vertical-align: top;"> <p>Enter password and user ID</p> <p>Select major function</p> <p>Other functions may also be selected</p> <p>Select surveillance</p> <p>Thumbnail views</p> <p>Select specific camera - thumbnails</p> <p>Select camera icon</p> <p>View camera output in labelled window</p> </td> <td style="vertical-align: top;"> <p>Generate video output</p> </td> <td style="vertical-align: top;"> <p>Valid passwords/ID</p> <p>Invalid passwords/ID</p> <p>Prompt for reentry</p> <p>Input tries remain</p> <p>No input tries remain</p> <p>Prompt for another view</p> <p>Exit this function</p> <p>See another camera</p> </td> </tr> </tbody> </table> </div>	Homeowner	Camera	Interface	<p>Enter password and user ID</p> <p>Select major function</p> <p>Other functions may also be selected</p> <p>Select surveillance</p> <p>Thumbnail views</p> <p>Select specific camera - thumbnails</p> <p>Select camera icon</p> <p>View camera output in labelled window</p>	<p>Generate video output</p>	<p>Valid passwords/ID</p> <p>Invalid passwords/ID</p> <p>Prompt for reentry</p> <p>Input tries remain</p> <p>No input tries remain</p> <p>Prompt for another view</p> <p>Exit this function</p> <p>See another camera</p>	10	CO1	L3
Homeowner	Camera	Interface								
<p>Enter password and user ID</p> <p>Select major function</p> <p>Other functions may also be selected</p> <p>Select surveillance</p> <p>Thumbnail views</p> <p>Select specific camera - thumbnails</p> <p>Select camera icon</p> <p>View camera output in labelled window</p>	<p>Generate video output</p>	<p>Valid passwords/ID</p> <p>Invalid passwords/ID</p> <p>Prompt for reentry</p> <p>Input tries remain</p> <p>No input tries remain</p> <p>Prompt for another view</p> <p>Exit this function</p> <p>See another camera</p>								
2a)	<p>Write a formal use case for surveillance.</p> <p>Use case activities: 3 Marks Explanation: 3 Marks</p>	6	CO1	L2						

	<p>SAFEHOME</p> <p><i>Use Case Template for Surveillance</i></p> <p>Use case: Access camera surveillance via the Internet—display camera views (ACS-DCV)</p> <p>Iteration: 2, last modification: January 14 by V. Roman.</p> <p>Primary actor: Homeowner.</p> <p>Goal in context: To view output of camera placed throughout the house from any remote location via the Internet. System must be fully configured; appropriate user ID and passwords must be obtained.</p> <p>Preconditions:</p> <p>Trigger: The homeowner decides to take a look inside the house while away.</p> <p>Scenario:</p> <ol style="list-style-type: none"> 1. The homeowner logs onto the SafeHome Products website. 2. The homeowner enters his or her user ID. 3. The homeowner enters two passwords (each at least eight characters in length). 4. The system displays all major function buttons. 5. The homeowner selects the "surveillance" from the major function buttons. 6. The homeowner selects "pick a camera." 7. The system displays the floor plan of the house. 8. The homeowner selects a camera icon from the floor plan. 9. The homeowner selects the "view" button. 10. The system displays a viewing window that is identified by the camera ID. 11. The system displays video output within the viewing window at one frame per second. <p>Exceptions:</p> <ol style="list-style-type: none"> 1. ID or passwords are incorrect or not recognized—see use case Validate ID and passwords. 2. Surveillance function not configured for this system—system displays appropriate error message; see use case Configure surveillance function. 3. Homeowner selects "View thumbnail snapshots for all camera"—see use case View thumbnail snapshots for all cameras. 4. A floor plan is not available or has not been configured—display appropriate error message and see use case Configure floor plan. 5. An alarm condition is encountered—see use case Alarm condition encountered. <p>Priority: Moderate priority, to be implemented after basic functions.</p> <p>When available: Third increment.</p> <p>Frequency of use: Moderate frequency.</p> <p>Channel to actor: Via PC-based browser and Internet connection.</p> <p>Secondary actors: System administrator, cameras.</p> <p>Channels to secondary actors:</p> <ol style="list-style-type: none"> 1. System administrator: PC-based system. 2. Cameras: wireless connectivity. <p>Open issues:</p> <ol style="list-style-type: none"> 1. What mechanisms protect unauthorized use of this capability by employees of SafeHome Products? 2. Is security sufficient? Hacking into this feature would represent a major invasion of privacy. 3. Will system response via the Internet be acceptable given the bandwidth required for camera views? 4. Will we develop a capability to provide video at a higher frames-per-second rate when high-bandwidth connections are available? 			
2b)	<p>Draw Preliminary use case diagram for Safe Home system.</p> <p>Use case diagram: 4 Marks</p> 	4	CO1	L2
3	<p>What is agility in the context of software engineering work? How you define change costs as a function of time in development.</p> <p>Agility: 5 Marks Change costs: 5 Marks</p> <p style="text-align: right;">Cont..</p> <p>what is agility in the context of software engineering work?</p> <p>Ivar Jacobson [Jac02a]:</p> <ul style="list-style-type: none"> ❖ <i>Agility</i> has become today's buzzword when describing a modern software process. Everyone is agile. ❖ An agile team is a nimble team able to appropriately respond to changes. Change is what software development is very much about. ✓ Changes in the software being built, ✓ Changes to the team members, ✓ Changes because of new technology, ✓ Changes of all kinds that may have an impact on the product they build or the project that creates the product. 	10	CO2	L2

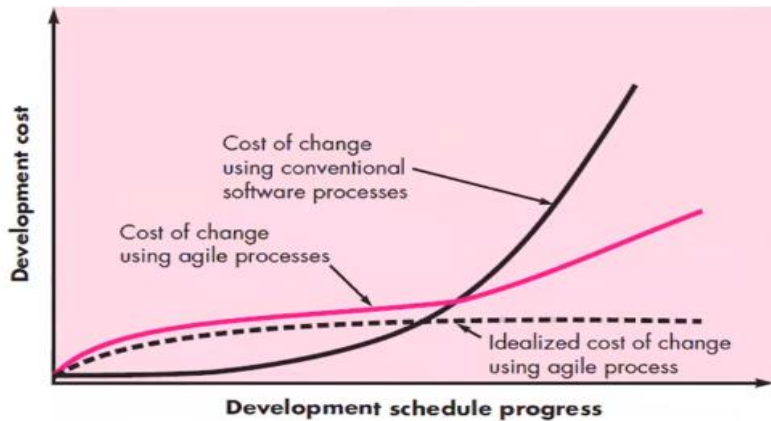


Figure: Change costs as a function of time in development

4

What is the difference between requirement modelling and requirement analysis. Write more about the requirement analysis with a diagram.

10

CO1

L2

Requirement modeling: 3 Marks

Requirement analysis: 3 Marks

Requirement analysis subtopics & diagram: 4 Marks

❖ Requirements modelling uses a combination of **text** and **diagrammatic** forms to depict requirements in a way that is relatively **easy to understand**, and more important, straightforward to review for **correctness, completeness, and consistency**.

• Requirements analysis results in the **specification of software's operational characteristics**, indicates **software's interface with other system elements**, and **establishes constraints** that software must meet.

- ✓ **Objectives** ✓
- ✓ **Analysis Rules of Thumb** ✓
- ✓ **Domain Analysis** ✓
- ✓ **Requirements Modeling Approaches**

The requirements model must achieve **three primary objectives**:

1. To **describe what the customer requires**
2. To **establish a basis for the creation of a software design**, and
3. To **define a set of requirements that can be validated once the software is built**.

Analysis rules of thumb

1. The model should focus on requirements that are visible within the problem or business domain. The level of abstraction should be relatively high.
2. Each element of the requirements model should add to an overall understanding of software requirements and provide insight into the information domain, function, and behavior of the system.
3. Delay consideration of infrastructure and other nonfunctional models until design.
4. Minimize coupling throughout the system.
5. Be certain that the requirements model provides value to all stakeholders.
6. Keep the model as simple as it can be.

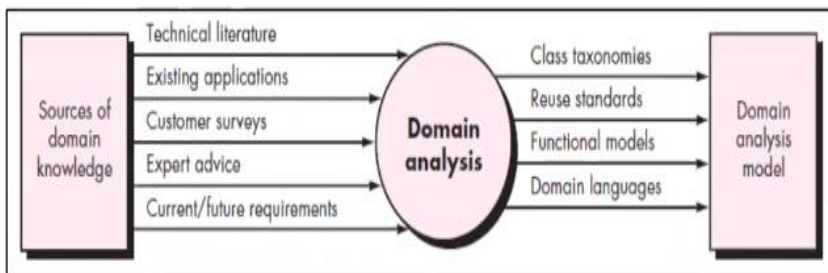
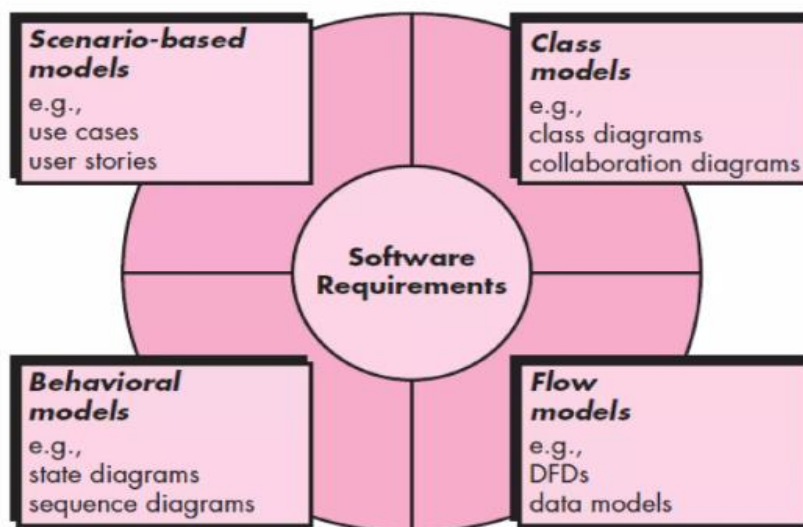


Figure. Input and output for domain analysis



5 Explain about the software engineering knowledge and core principles.

10

CO2

L1

Software engineering knowledge: 5 Marks

Core principles: 5 Marks

In an editorial in *IEEE Software*, Steve McConnell commented:

Many software practitioners think of software engineering knowledge almost exclusively as knowledge of specific technologies: Java, Perl, html, C++, Linux, Windows NT, and so on. Software development knowledge has a 3-year half-life:

But knowledge of "software engineering principles" will serve a professional programmer throughout his career.

McConnell says software engineering knowledge had evolved to a "stable core" is "75 percent of the knowledge needed to develop a complex system."

The following set of core principles can be applied to the framework, and to every software process.

Principle 1. Be agile.

Whether the process model you choose is prescriptive or agile, keep your technical approach as simple as possible, work products as concise as possible, and make decisions locally whenever possible.

Principle 2. Focus on quality at every step. The exit condition for every process activity, action, and task should focus on the quality of the work product produced.

Principle 3. Be ready to adapt. Process is not religion experience, so, adapt your approach to constraints imposed by the problem, the people, and the project itself.

Principle 4. Build an effective team. Bottom line is people. Build a self-organizing team that has mutual trust and respect.

Principle 5. Establish mechanisms for communication and coordination.

Projects fail because important information falls into the cracks and/or stakeholders fail to coordinate their efforts to create a successful end product.

Principle 6. Manage change. Either formal or informal approach, but mechanisms must be established to manage the way changes are requested, assessed, approved, and implemented.

Principle 7. Assess risk. Lots of things can go wrong as software is developed. Establish contingency plans.

Principle 8. Create work products that provide value for others. Be sure that the work product imparts the necessary information without ambiguity or omission.

A list of required functions and features will be passed along to the person (people) who will develop a design. The design will be passed along to those who generate code, and so on.

6 Explain about the Principles that guide each framework activity in detail.

10

CO2

L1

10 Principles: Each 1 Mark

Principle 1. Listen.

Focus on the speaker's words, rather than formulating your response.

Ask for clarification, but avoid constant interruptions.

Never become argumentative (e.g., rolling your eyes or shaking your head).

Principle 2. Prepare before you communicate.

Do some research to understand business domain terminology.



Principle 3. Someone should facilitate the activity.

Every communication meeting should have a leader (a facilitator) to keep conversation in a productive direction, and to mediate for any conflict.

Principle 4. Face-to-face communication is best.

Usually works better when some other representation is present.

For example, create a drawing or a "strawman" document that serves as a focus for discussion.

Principle 5. Take notes and document decisions.



Someone should serve as a "recorder" and note all important points and decisions.

Principle 6. Strive for collaboration.

Each small collaboration serves to build trust among team members and creates a common goal.

Principle 7. Stay focused; modularize your discussion.

The facilitator should keep the conversation modular, leaving one topic only after it has been resolved.

Principle 8. If something is unclear, draw a picture.

A sketch or drawing often provide clarity when words fail to do the job.

Principle 9.

(a) Once you agree to something, move on.

(b) If you can't agree to something, move on.

(c) If a feature or function is unclear and cannot be clarified at the moment, move on.

Many topics require iterative discussion and that "moving on" is sometimes the best way to achieve communication agility.

Principle 10. Negotiation is not a contest or a game. It works best when both parties win.

When we negotiate functions and features, priorities, and delivery, it demands compromise from all parties.

