

MODULE - 1

- 1 Define Software ? Explain briefly the characteristics of the software
- A Software system usually consists of a number of separate programs, configuration files, system documentation, and user documentation.

Characteristics of the Software

- ① Maintainability - Software must evolve to meet changing needs
- ② Dependability - Software must be trustworthy, reliable
- ③ Efficiency - Software should not make wasteful use of system resources
- ④ Usability - Software are must be usable by the users for which it was designed .

There are two fundamental types of software product

- ① Generic products: These are stand-alone systems that are produced by a development organization and sold on the open market to any customer who is able to buy them.
Eg: MS Word, Windows XP etc

- ② Customized (or bespoke): These are systems which are commissioned by a particular customer. A software contractor develops the software especially for that customer.

Eg: Bank management, office automation etc

8 Define Software process and explain its typical framework and umbrella activities

A Software process :

- A software process is a set of activities that lead to the production of a software product. These activities may involve the development of software from scratch or extending and modifying existing systems and by configuring and integrating off-the-shelf software or system components.

Framework

- ① Software Specification : The functionality of the software and constraints on its operation must be defined.
- ② Software design and implementation : The software to meet the specification must be produced
- ③ Software Validation : The software must be validated to ensure that it does what the customer wants
- ④ Software evolution : The software must evolve to meet changing customer needs.

Umbrella Activities

- ① Software project tracking and control
 - Allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.
- ② Risk management
 - Assesses risks that may affect the outcome of the project or the quality of product
- ③ Software quality assurance
 - Defines and conducts the activities required to ensure software quality.

④ Technical reviews

Assesses software engineering work products in an effort to uncover & remove errors before they are propagated to the next activity

⑤ Measurement

Defines and collects process, project and product measures that assist the team in delivering software that meets stakeholders needs; can be used in conjunction with all other framework and umbrella activities.

⑥ Software configuration management

Manages the effects of change throughout the software process

⑦ Reusability management

Defines criteria for work product reuse and establishes mechanisms to achieve reusable components

⑧ Work product preparation and production

Encompasses the activities required to create work products such as models, documents, logs, forms, and lists

3 Define Software Engineering? Explain the essence of Software engineering practice and principles

A Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification to maintaining the system after it has gone into use.

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software

Practices

- ① Procedures : defines what are the activities and when they will be executed bind the methods and tools into a framework
- ② Methods : defines how each activity is executed using the technology provide the rules and steps for carrying out software engineering tasks, such as project planning, requirements analysis, design, coding, testing & maintenance
- ③ Tools : Provide automated or semi-automated support for methods. Tools that automates a range of software engineering methods, called Computer-Aided Software Engineering (CASE)

Principles

- ① Public - Software engineers shall act consistently with the public interest
- ② Client and employer - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest
- ③ Product - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible
- ④ Judgment - software engineers shall maintain integrity and independence in their professional judgement
- ⑤ Management - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development & maintenance
- ⑥ Profession - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest

- ④ Colleagues - Software engineers shall be fair to and supportive of their colleagues.
- ⑤ Self - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to their the practice of the profession

4 What is Webapp? Explain the various attributes of Webapp

A → A webapp is a software application that runs on a web server and is accessed through a web browser over the internet or an intranet. Unlike traditional desktop applications, webapps do not require installation on a local device; they rely on a browser interface to interact with the user

Attributes of webapp

① Network-intensive

→ A webapp relies on a network. It can only function when users are connected

② Content-Driven

Content (text, images, audio, video, documents) is a major part of most webapps

③ Continuous Evolution

Webapps evolve constantly. Developers update content, add new features enhance UI and fix bugs more frequently than traditional software

④ Immediacy

Updates are immediately available to users since the app runs on the web server

⑤ Security

User Security is a crucial attribute because webapps are

exposed to attacks like hacking, phishing and data theft.

⑥ Scalability and Performance

Webapps must handle a large and variable number of users yet efficiently.

5 list and describe the various software application domains

1 ① System Software:

- System software provides the basic functions for computer operation and acts as a platform for application software
 - Purpose : Manages hardware resources, file systems, memory and enables communication b/w hardware and user applications
- Eg: operating systems

② Application Software

Programs designed to perform specific tasks for the end users

Purpose : Helps users complete daily tasks such as document creation, design, communication etc

Eg: Microsoft office

③ Web applications

Applications accessed through web browsers over the internet

Purpose: provide interactive services such as communication, content sharing and online collaboration

Eg: Gmail, youtube

④ Embedded Software

Software built into devices to control their specific functions

Purpose: control hardware devices and enable them to perform dedicated tasks efficiently.

Eg: Software in washing machines, SmartTVs

⑤ Mobile applications

Software designed for smartphones and tablets

purpose: provide communication, navigation, entertainment, and utility services on mobile devices

Eg: WhatsApp, Instagram

⑥ Real-time Software

Software that responds to inputs immediately, within strict time constraints

purpose: Ensure quick and predictable response for critical operations

Eg: Air traffic control, robotics.

⑥ Explain the Various tasks in requirement engineering

A. ① Requirement Elicitation [Gathering]

This is the process of collecting requirements from stakeholders

Techniques: Interviews, workshops, observations, use cases

Goal: Understand what the stakeholders need from the system

② Requirement Analysis

In this task, gathered requirements are examined for feasibility, clarity and conflicts.

Conflicting requirements are identified and resolved

Goal: Ensure requirements are consistent, complete and realistic

③ Requirement Specification

Writing down the analyzed requirements in a clear and structured format

Typically documented in software requirement specification (SRS) document

④ Requirement Validation

Checking that the documented requirements truly reflect stakeholder's needs.

Goal: Detect errors early

⑤ Requirement Management

Continuous process of handling changes in requirements during the software lifecycle

Helps adapt to evolving business needs, market changes or regulatory updates

Goal: keep requirements up-to-date and aligned with project process

List and explain the different steps involved in establishing the ground work

A ① Feasibility Study

- Assesses whether the project is technically, economically and operationally feasible
- Types:
 - Technical feasibility - Can it be built with current tech?
 - Economic feasibility - Is it cost-effective?
 - Operational feasibility - Will it function in the target environment

② Stakeholder Identification

- Recognize all people who affect or are affected by the system
- Types
 - Internal (developers, testers)
 - External (clients, users, regulators)
- Stakeholder map / chart can be created.

③ Business Case definition

- Why the project is needed
- Key drivers (cost reduction, automation, compliance)
- High level goals and expected benefits

④ Constraints identification

Budget, deadlines, technology limits, regulatory/legal requirements

8 What is Requirements Elicitation and illustrate this process with reference to the safehome project

* The process of gathering requirements from stakeholders, users, and other sources.

① Identify stakeholders

- House owners
- Security company
- Installer team
- Emergency services

② Gather requirements

- Interviews with homeowners
 - Need remote access through mobile app
 - Want motion detection & alarms
- Observation
 - Current system is difficult to use, they forgot to arm it
- Workshops with security staff
 - Need real-time alerts if an intrusion occurs
- Questionnaires for broader customers
 - perform for preference for voice alert alerts, panic button

③ Refine & Document requirement

- functional requirements: The system shall detect unauthorised entry using motion sensors
- Non-functional requirements: The mobile app must support android & IOS

④ Resolve Conflicts

Some homeowners want automatic arming others find it inconvenient

⑤ Validate requirements

Get user feedback & confirm requirements before moving to design

9 Explain with example how to develop use cases

A: Usecase Name: Place an order

① Actor : customer

② Preconditions :

- Customer must be logged into website
- Customer has items in the Cart

③ Scenario

- Customer goes to Cart
- Customer clicks "checkout"
- System asks for delivery address & payment details
- Customer provides details
- System processes the payment
- System confirms the order

④ Postconditions .

- Order is placed successfully
- Customer receives confirmation.

10 Explain elements required to build a requirements model.

A: ① Scenario - Based modeling

Captures how users interact with the system in real-world scenarios

Tools : Usecases Diagrams (UML)

Elements :

- Actors (users/ systems)
- Usecases (functions)

- Relationships (extends / includes)

Example:

- Actor : customer
- Usecase : Place order, Track order, Cancel order

② Class-Based modeling

represents data structure, objects and relationships

- Tools: class diagrams (UML)
- Elements:

- classes with attributes & methods
- Relationships

Example:

Class: Customer

→ Attributes: name, phone no

→ Methods: login(), place order()

③ Flow-oriented modeling

Shows the flow of data and transformations across the system

Tool: Data flow diagram (DFD)

Elements:

- Processes
- Datastores
- Data flows
- External entities

Example

Process: "Process order"

Datastore: "Customer info"

External Entity: "Payment gateway"

(4) Behavioral modeling

Models system behavior in response to events over time

Tools/elements:

- State diagrams - show system states & transitions
- Activity diagrams - model workflows and control flows

Example

Order state transitions: New \rightarrow Confirmed \rightarrow Shipped \rightarrow Delivered.

MODULE - 1

- 1 Define Software ? Explain briefly the characteristics of the software
- A Software system usually consists of a number of separate programs, configuration files, system documentation, and user documentation.

Characteristics of the Software

- ① Maintainability - Software must evolve to meet changing needs
- ② Dependability - Software must be trustworthy, reliable
- ③ Efficiency - Software should not make wasteful use of system resources
- ④ Usability - Software are must be usable by the users for which it was designed .

There are two fundamental types of software product

- ① Generic products: These are stand-alone systems that are produced by a development organization and sold on the open market to any customer who is able to buy them.
Eg: MS Word, Windows XP etc

- ② Customized (or bespoke): These are systems which are commissioned by a particular customer. A software contractor develops the software especially for that customer.

Eg: Bank management, office automation etc

8 Define Software process and explain its typical framework and umbrella activities

A Software process :

- A software process is a set of activities that lead to the production of a software product. These activities may involve the development of software from scratch or extending and modifying existing systems and by configuring and integrating off-the-shelf software or system components.

Framework

- ① Software Specification : The functionality of the software and constraints on its operation must be defined.
- ② Software design and implementation : The software to meet the specification must be produced
- ③ Software Validation : The software must be validated to ensure that it does what the customer wants
- ④ Software evolution : The software must evolve to meet changing customer needs.

Umbrella Activities

- ① Software project tracking and control
 - Allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.
- ② Risk management
 - Assesses risks that may affect the outcome of the project or the quality of product
- ③ Software quality assurance
 - Defines and conducts the activities required to ensure software quality.

④ Technical reviews

Assesses software engineering work products in an effort to uncover & remove errors before they are propagated to the next activity

⑤ Measurement

Defines and collects process, project and product measures that assist the team in delivering software that meets stakeholders needs; can be used in conjunction with all other framework and umbrella activities.

⑥ Software configuration management

Manages the effects of change throughout the software process

⑦ Reusability management

Defines criteria for work product reuse and establishes mechanisms to achieve reusable components

⑧ Work product preparation and production

Encompasses the activities required to create work products such as models, documents, logs, forms, and lists

3 Define Software Engineering? Explain the essence of Software engineering practice and principles

A Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification to maintaining the system after it has gone into use.

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software

Practices

- ① Procedures : defines what are the activities and when they will be executed bind the methods and tools into a framework
- ② Methods : defines how each activity is executed using the technology provide the rules and steps for carrying out software engineering tasks, such as project planning, requirements analysis, design, coding, testing & maintenance
- ③ Tools : Provide automated or semi-automated support for methods. Tools that automates a range of software engineering methods, called Computer-Aided Software Engineering (CASE)

Principles

- ① Public - Software engineers shall act consistently with the public interest
- ② Client and employer - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest
- ③ Product - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible
- ④ Judgment - software engineers shall maintain integrity and independence in their professional judgement
- ⑤ Management - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development & maintenance
- ⑥ Profession - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest

- ④ Colleagues - Software engineers shall be fair to and supportive of their colleagues.
- ⑤ Self - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to their the practice of the profession

4 What is Webapp? Explain the various attributes of Webapp

A → A webapp is a software application that runs on a web server and is accessed through a web browser over the internet or an intranet. Unlike traditional desktop applications, webapps do not require installation on a local device; they rely on a browser interface to interact with the user

Attributes of webapp

① Network-intensive

→ A webapp relies on a network. It can only function when users are connected

② Content-Driven

Content (text, images, audio, video, documents) is a major part of most webapps

③ Continuous Evolution

Webapps evolve constantly. Developers update content, add new features enhance UI and fix bugs more frequently than traditional software

④ Immediacy

Updates are immediately available to users since the app runs on the web server

⑤ Security

User Security is a crucial attribute because webapps are

exposed to attacks like hacking, phishing and data theft.

⑥ Scalability and Performance

Webapps must handle a large and variable number of users yet efficiently.

5 list and describe the various software application domains

1 ① System Software:

- System software provides the basic functions for computer operation and acts as a platform for application software
 - Purpose : Manages hardware resources, file systems, memory and enables communication b/w hardware and user applications
- Eg: operating systems

② Application Software

Programs designed to perform specific tasks for the end users

Purpose : Helps users complete daily tasks such as document creation, design, communication etc

Eg: Microsoft office

③ Web applications

Applications accessed through web browsers over the internet

Purpose: provide interactive services such as communication, content sharing and online collaboration

Eg: Gmail, youtube

④ Embedded Software

Software built into devices to control their specific functions

Purpose: control hardware devices and enable them to perform dedicated tasks efficiently.

Eg: Software in washing machines, SmartTVs

⑤ Mobile applications

Software designed for smartphones and tablets

purpose: provide communication, navigation, entertainment, and utility services on mobile devices

Eg: WhatsApp, Instagram

⑥ Real-time Software

Software that responds to inputs immediately, within strict time constraints

purpose: Ensure quick and predictable response for critical operations

Eg: Air traffic control, robotics.

⑥ Explain the Various tasks in requirement engineering

A. ① Requirement Elicitation [Gathering]

This is the process of collecting requirements from stakeholders

Techniques: Interviews, workshops, observations, use cases

Goal: Understand what the stakeholders need from the system

② Requirement Analysis

In this task, gathered requirements are examined for feasibility, clarity and conflicts.

Conflicting requirements are identified and resolved

Goal: Ensure requirements are consistent, complete and realistic

③ Requirement Specification

Writing down the analyzed requirements in a clear and structured format

Typically documented in software requirement specification (SRS) document

④ Requirement Validation

Checking that the documented requirements truly reflect stakeholder's needs.

Goal: Detect errors early

⑤ Requirement Management

Continuous process of handling changes in requirements during the software lifecycle

Helps adapt to evolving business needs, market changes or regulatory updates

Goal: keep requirements up-to-date and aligned with project process

List and explain the different steps involved in establishing the ground work

A ① Feasibility Study

- Assesses whether the project is technically, economically and operationally feasible
- Types:
 - Technical feasibility - Can it be built with current tech?
 - Economic feasibility - Is it cost-effective?
 - Operational feasibility - Will it function in the target environment

② Stakeholder Identification

- Recognize all people who affect or are affected by the system
- Types
 - Internal (developers, testers)
 - External (clients, users, regulators)
- Stakeholder map / chart can be created.

③ Business Case definition

- Why the project is needed
- Key drivers (cost reduction, automation, compliance)
- High level goals and expected benefits

④ Constraints identification

Budget, deadlines, technology limits, regulatory/legal requirements

8 What is Requirements Elicitation and illustrate this process with reference to the safehome project

* The process of gathering requirements from stakeholders, users, and other sources.

① Identify stakeholders

- House owners
- Security company
- Installer team
- Emergency Services

② Gather requirements

- Interviews with homeowners
 - Need remote access through mobile app
 - Want motion detection & alarms
- Observation
 - Current system is difficult to use, they forgot to arm it
- Workshops with security staff
 - Need real-time alerts if an intrusion occurs
- Questionnaires for broader customers
 - perform for preference for voice alert alerts, panic button

③ Refine & Document requirement

- functional requirements: The system shall detect unauthorised entry using motion sensors
- Non-functional requirements: The mobile app must support android & IOS

④ Resolve Conflicts

Some homeowners want automatic arming others find it inconvenient

⑤ Validate requirements

Get user feedback & confirm requirements before moving to design

9 Explain with example how to develop use cases

A: Usecase Name: Place an order

① Actor : customer

② Preconditions :

- Customer must be logged into website
- Customer has items in the cart

③ Scenario

- Customer goes to cart
- Customer clicks "checkout"
- System asks for delivery address & payment details
- Customer provides details
- System processes the payment
- System confirms the order

④ Postconditions.

- Order is placed successfully
- Customer receives confirmation.

10 Explain elements required to build a requirements model.

A: ① Scenario - Based modeling

Captures how users interact with the system in real-world scenarios

Tools: Usecases Diagrams (UML)

Elements:

- Actors (users/ systems)
- Usecases (functions)

- Relationships (extends / includes)

Example:

- Actor : customer
- Usecase : Place order, Track order, Cancel order

② Class-Based modeling

represents data structure, objects and relationships

- Tools: class diagrams (UML)
- Elements:

- classes with attributes & methods
- Relationships

Example:

Class: Customer

→ Attributes: name, phone no

→ Methods: login(), place order()

③ Flow-oriented modeling

Shows the flow of data and transformations across the system

Tool: Data flow diagram (DFD)

Elements:

- Processes
- Datastores
- Data flows
- External entities

Example

Process: "Process order"

Datastore: "Customer info"

External Entity: "Payment gateway"

(4) Behavioral modeling

Models system behavior in response to events over time

Tools/elements:

- State diagrams - show system states & transitions
- Activity diagrams - model workflows and control flows

Example

Order state transitions: New \rightarrow Confirmed \rightarrow Shipped \rightarrow Delivered.

MODULE - 1

- 1 Define Software ? Explain briefly the characteristics of the software
- A Software system usually consists of a number of separate programs, configuration files, system documentation, and user documentation.

Characteristics of the Software

- ① Maintainability - Software must evolve to meet changing needs
- ② Dependability - Software must be trustworthy, reliable
- ③ Efficiency - Software should not make wasteful use of system resources
- ④ Usability - Software are must be usable by the users for which it was designed .

There are two fundamental types of software product

- ① Generic products: These are stand-alone systems that are produced by a development organization and sold on the open market to any customer who is able to buy them.
Eg: MS Word, Windows XP etc

- ② Customized (or bespoke): These are systems which are commissioned by a particular customer. A software contractor develops the software especially for that customer.

Eg: Bank management, office automation etc

8 Define Software process and explain its typical framework and umbrella activities

A Software process :

- A software process is a set of activities that lead to the production of a software product. These activities may involve the development of software from scratch or extending and modifying existing systems and by configuring and integrating off-the-shelf software or system components.

Framework

- ① Software Specification : The functionality of the software and constraints on its operation must be defined.
- ② Software design and implementation : The software to meet the specification must be produced
- ③ Software Validation : The software must be validated to ensure that it does what the customer wants
- ④ Software evolution : The software must evolve to meet changing customer needs.

Umbrella Activities

- ① Software project tracking and control
 - Allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.
- ② Risk management
 - Assesses risks that may affect the outcome of the project or the quality of product
- ③ Software quality assurance
 - Defines and conducts the activities required to ensure software quality.

④ Technical reviews

Assesses software engineering work products in an effort to uncover & remove errors before they are propagated to the next activity

⑤ Measurement

Defines and collects process, project and product measures that assist the team in delivering software that meets stakeholders needs; can be used in conjunction with all other framework and umbrella activities.

⑥ Software configuration management

Manages the effects of change throughout the software process

⑦ Reusability management

Defines criteria for work product reuse and establishes mechanisms to achieve reusable components

⑧ Work product preparation and production

Encompasses the activities required to create work products such as models, documents, logs, forms, and lists

3 Define Software Engineering? Explain the essence of Software engineering practice and principles

A Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification to maintaining the system after it has gone into use.

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software

Practices

- ① Procedures : defines what are the activities and when they will be executed bind the methods and tools into a framework
- ② Methods : defines how each activity is executed using the technology provide the rules and steps for carrying out software engineering tasks, such as project planning, requirements analysis, design, coding, testing & maintenance
- ③ Tools : Provide automated or semi-automated support for methods. Tools that automates a range of software engineering methods, called Computer-Aided Software Engineering (CASE)

Principles

- ① Public - Software engineers shall act consistently with the public interest
- ② Client and employer - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest
- ③ Product - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible
- ④ Judgment - software engineers shall maintain integrity and independence in their professional judgement
- ⑤ Management - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development & maintenance
- ⑥ Profession - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest

- ④ Colleagues - Software engineers shall be fair to and supportive of their colleagues.
- ⑤ Self - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to their the practice of the profession

4 What is Webapp? Explain the various attributes of Webapp

A → A webapp is a software application that runs on a web server and is accessed through a web browser over the internet or an intranet. Unlike traditional desktop applications, webapps do not require installation on a local device; they rely on a browser interface to interact with the user

Attributes of webapp

① Network-intensive

→ A webapp relies on a network. It can only function when users are connected

② Content-Driven

Content (text, images, audio, video, documents) is a major part of most webapps

③ Continuous Evolution

Webapps evolve constantly. Developers update content, add new features enhance UI and fix bugs more frequently than traditional software

④ Immediacy

Updates are immediately available to users since the app runs on the web server

⑤ Security

User Security is a crucial attribute because webapps are

exposed to attacks like hacking, phishing and data theft.

⑥ Scalability and Performance

Webapps must handle a large and variable number of users yet efficiently.

5 list and describe the various software application domains

1 ① System Software:

- System software provides the basic functions for computer operation and acts as a platform for application software
 - Purpose : Manages hardware resources, file systems, memory and enables communication b/w hardware and user applications
- Eg: operating systems

② Application Software

Programs designed to perform specific tasks for the end users

Purpose : Helps users complete daily tasks such as document creation, design, communication etc

Eg: Microsoft office

③ Web applications

Applications accessed through web browsers over the internet

Purpose: provide interactive services such as communication, content sharing and online collaboration

Eg: Gmail, youtube

④ Embedded Software

Software built into devices to control their specific functions

Purpose: control hardware devices and enable them to perform dedicated tasks efficiently.

Eg: Software in washing machines, SmartTVs

⑤ Mobile applications

Software designed for smartphones and tablets

purpose: provide communication, navigation, entertainment, and utility services on mobile devices

Eg: WhatsApp, Instagram

⑥ Real-time Software

Software that responds to inputs immediately, within strict time constraints

purpose: Ensure quick and predictable response for critical operations

Eg: Air traffic control, robotics.

⑥ Explain the Various tasks in requirement engineering

A. ① Requirement Elicitation [Gathering]

This is the process of collecting requirements from stakeholders

Techniques: Interviews, workshops, observations, use cases

Goal: Understand what the stakeholders need from the system

② Requirement Analysis

In this task, gathered requirements are examined for feasibility, clarity and conflicts.

Conflicting requirements are identified and resolved

Goal: Ensure requirements are consistent, complete and realistic

③ Requirement Specification

Writing down the analyzed requirements in a clear and structured format

Typically documented in software requirement specification (SRS) document

④ Requirement Validation

Checking that the documented requirements truly reflect stakeholder's needs.

Goal: Detect errors early

⑤ Requirement Management

Continuous process of handling changes in requirements during the software lifecycle

Helps adapt to evolving business needs, market changes or regulatory updates

Goal: keep requirements up-to-date and aligned with project process

List and explain the different steps involved in establishing the ground work

A ① Feasibility Study

- Assesses whether the project is technically, economically and operationally feasible
- Types:
 - Technical feasibility - Can it be built with current tech?
 - Economic feasibility - Is it cost-effective?
 - Operational feasibility - Will it function in the target environment

② Stakeholder Identification

- Recognize all people who affect or are affected by the system
- Types
 - Internal (developers, testers)
 - External (clients, users, regulators)
- Stakeholder map / chart can be created.

③ Business Case definition

- Why the project is needed
- Key drivers (cost reduction, automation, compliance)
- High level goals and expected benefits

④ Constraints identification

Budget, deadlines, technology limits, regulatory/legal requirements

8 What is Requirements Elicitation and illustrate this process with reference to the safehome project

* The process of gathering requirements from stakeholders, users, and other sources.

① Identify stakeholders

- House owners
- Security company
- Installer team
- Emergency Services

② Gather requirements

- Interviews with homeowners
 - Need remote access through mobile app
 - Want motion detection & alarms
- Observation
 - Current system is difficult to use, they forgot to arm it
- Workshops with security staff
 - Need real-time alerts if an intrusion occurs
- Questionnaires for broader customers
 - perform for preference for voice alert alerts, panic button

③ Refine & Document requirement

- functional requirements: The system shall detect unauthorised entry using motion sensors
- Non-functional requirements: The mobile app must support android & IOS

④ Resolve Conflicts

Some homeowners want automatic arming others find it inconvenient

⑤ Validate requirements

Get user feedback & confirm requirements before moving to design

9 Explain with example how to develop use cases

A: Usecase Name: Place an order

① Actor : customer

② Preconditions :

- Customer must be logged into website
- Customer has items in the Cart

③ Scenario

- Customer goes to Cart
- Customer clicks "checkout"
- System asks for delivery address & payment details
- Customer provides details
- System processes the payment
- System confirms the order

④ Postconditions .

- Order is placed successfully
- Customer receives confirmation.

10 Explain elements required to build a requirements model.

A: ① Scenario - Based modeling

Captures how users interact with the system in real-world scenarios

Tools : Usecases Diagrams (UML)

Elements :

- Actors (users/ systems)
- Usecases (functions)

- Relationships (extends / includes)

Example:

- Actor : customer
- Usecase : Place order, Track order, Cancel order

② Class-Based modeling

represents data structure, objects and relationships

- Tools: class diagrams (UML)
- Elements:

- classes with attributes & methods
- Relationships

Example:

Class: Customer

→ Attributes: name, phone no

→ Methods: login(), place order()

③ Flow-oriented modeling

Shows the flow of data and transformations across the system

Tool: Data flow diagram (DFD)

Elements:

- Processes
- Datastores
- Data flows
- External entities

Example

Process: "Process order"

Datastore: "Customer info"

External Entity: "Payment gateway"

(4) Behavioral modeling

Models system behavior in response to events over time

Tools/elements:

- State diagrams - show system states & transitions
- Activity diagrams - model workflows and control flows

Example

Order state transitions: New \rightarrow Confirmed \rightarrow Shipped \rightarrow Delivered.