


Unit-wise questions - Software Engineering

[Syllabus](#)[Notes](#)[Old Questions & solutions](#)[Yearly Questions](#)[Text & reference books](#)[Software Engineering](#)[Compiler Design and Construction](#)[E-Governance](#)[NET Centric Computing](#)[Technical Writing](#)[Applied Logic](#)[E-commerce](#)[Automation and Robotics](#)[Neural Networks](#)[Computer Hardware Design](#)[Cognitive Science](#)[Real Time System \(old-course\)](#)[Unit: 1 Introduction](#)[24 questions](#)[Unit: 2 Software Processes](#)[25 questions](#)[Unit: 3 Agile Software Development](#)[questions](#)[Unit: 4 Requirements Engineering](#)[24 questions](#)

3. What are the important activities that are carried out during the feasibility study phase? Explain.

 **hide solution**
asked in 2068(II)

Solution

Feasibility study is so important stage software life cycle as after completion of feasibility study it gives a conclusion of whether to go ahead with proposed project as it is practically feasible or to stop proposed project here as it is not right/feasible to develop or to think/analyze about proposed project again.

Along with this Feasibility study helps in identifying risk factors involved in developing and deploying system and planning for risk analysis also narrows the business alternatives and enhance success rate analyzing different parameters associated with proposed project development.

Types of feasibility study:

1. Technical Feasibility: Technical feasibility helps in accessing the current resources as well as technology that is required for accomplishing the requirements of the user in the software within the allocated budget and time. Following tasks are performed in the technical feasibility:


- Helps in determining whether the technology used is stable.
- Is the technology being demanded available? Or can the features being requested are possible to embed with the current/ available technology.
- Analyze the technical capabilities and skills of the team members of software development.

2. Operational/ Behavioral Feasibility: Behavioral feasibility is studied in order to check, whether the human or employees in the business will use it or not. Operational feasibility relies on human resources and analyzes whether the software will operate after it is developed properly or not.

3. Economic Feasibility: Economic Feasibility helps in determining whether the required software has the potential to generate financial gains for an organization. This type of study involves the cost incurred on the team of the software development, cost of study involved in conducting a feasibility study, estimated cost of software and hardware.

4. Scheduling feasibility: It is the most important study that is used for measuring the success of the project. All the projects will fail, if not completed at specified time. In scheduling feasibility, an organization is able to estimate the time required for completing the project. The time that will be required for the development of the system, is estimated on the basis of the requirements. Once all these things are sorted, manpower is allocated their respective task and efforts are made to complete the task within the deadline.

3. What are the major tasks of requirements engineering activity? Explain.

 **hide solution**
asked in 2073

Solution

Requirement engineering is the process of defining, documenting and maintaining the requirements in the engineering design process. The requirement engineering process consists of the following activities:

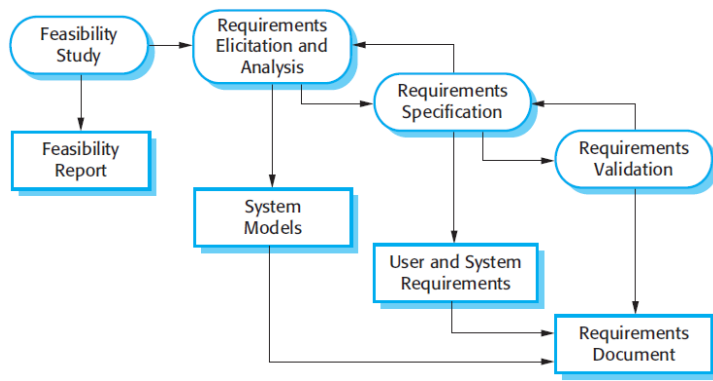


Fig: The requirements engineering process.

- 1. Feasibility study:** An estimate is made of whether the identified can be achieved using the current software and hardware technologies, under the current budget, etc. The feasibility study should be cheap and quick; it should inform the decision of whether or not to go ahead with the project.
- 2. Requirements elicitation and analysis:** This is the process of deriving the system requirements through observation of existing systems, discussions with stakeholders, etc. This may involve the development of one or more system models and prototypes that can help us understanding the system to be specified.
- 3. Requirements specification:** It's the activity of writing down the information gathered during the elicitation and analysis activity into a document that defines a set of requirements. Two types of requirements may be included in this document; user and system requirements.
- 4. Requirements validation:** It's the process of checking the requirements for realism, consistency and completeness. During this process, our goal is to discover errors in the requirements document. When errors are found, it must be modified to correct these problems.

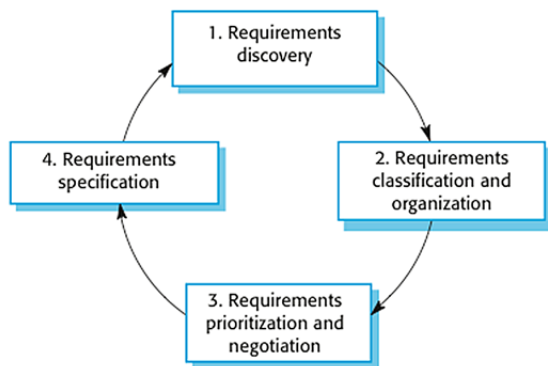
4. Discuss requirements elicitation and analysis activity of requirements engineering process.

hide solution
asked in 2071

Solution

Requirements elicitation and analysis is a process of interacting with customers and end-users to find out about the domain requirements, what services the system should provide, and the other constrains.

The requirements elicitation and analysis has 4 main process:



- 1. Requirements Discovery:** It's the process of interacting with, and gathering the requirements from, the stakeholders about the required system and the existing system (if exist). It can be done using some techniques, like interviews, scenarios, prototypes, etc, which help the stockholders to understand what the system will be like.
- 2. Requirements Classification & Organization:** It's very important to organize the overall structure of the system. Putting related requirements together, and decomposing the system into sub components of related requirements. Then, we define the relationship between these components.
- 3. Requirements Prioritization & Negotiation:** This activity is concerned with prioritizing requirements and finding and resolving requirements conflicts through negotiations until you reach a situation where some of the stakeholders can compromise.
- 4. Requirements Specification:** It is the process of writing down user and system requirements into a document. The requirement should be clear, easy to understand, complete and consistent.

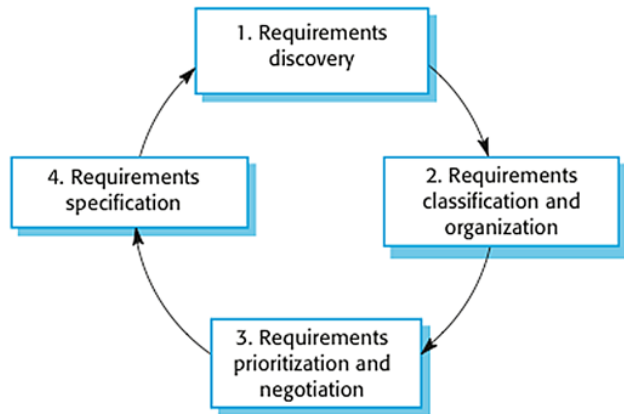
4. What is requirements elicitation and analysis? Discuss.

hide solution
asked in 2072

Solution

Requirements elicitation and analysis is a process of interacting with customers and end-users to find out about the domain requirements, what services the system should provide, and the other constraints.

The requirements elicitation and analysis has 4 main process:



1. Requirements Discovery: It's the process of interacting with, and gathering the requirements from, the stakeholders about the required system and the existing system (if exist). It can be done using some techniques, like interviews, scenarios, prototypes, etc, which help the stockholders to understand what the system will be like.

2. Requirements Classification & Organization: It's very important to organize the overall structure of the system. Putting related requirements together, and decomposing the system into sub components of related requirements. Then, we define the relationship between these components.

3. Requirements Prioritization & Negotiation: This activity is concerned with prioritizing requirements and finding and resolving requirements conflicts through negotiations until you reach a situation where some of the stakeholders can compromise.

4. Requirements Specification: It is the process of writing down user and system requirements into a document. The requirement should be clear, easy to understand, complete and consistent.

4. What is a software requirement? Explain software requirement process.

hide solution
asked in 2074

Solution

The requirement for a system are the descriptions of what the system should do - the services that it provides and the constraints on its operation. Requirement may range from a high-level abstract statement of a services or of a system constraint to detailed mathematical specification.

Software requirement process:

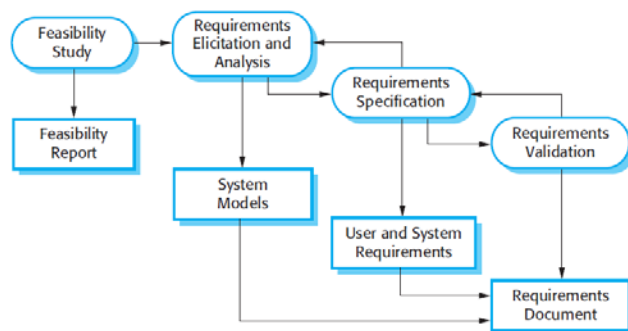


Fig: The requirements engineering process.


1. Feasibility study: An estimate is made of whether the identified can be achieved using the current software and hardware technologies, under the current budget, etc. The feasibility study should be cheap and quick; it should inform the decision of whether or not to go ahead with the project.

2. Requirements elicitation and analysis: This is the process of deriving the system requirements through observation of existing systems, discussions with stakeholders, etc. This may involve the development of one or more system models and prototypes that can help us understanding the system to be specified.

3. Requirements specification: It's the activity of writing down the information gathered during the elicitation and analysis activity into a document that defines a set of requirements. Two types of requirements may be included in this document; user and system requirements.

4. Requirements validation: It's the process of checking the requirements for realism, consistency and completeness. During this process, our goal is to discover errors in the requirements document. When errors are found, it must be modified to correct these problems.

4. Briefly explain functional, non-functional, and domain requirements.

 **hide solution**
asked in 2076

Solution

Functional Requirements

Functional requirements are the statements of services the system should provide, how the system should react to particular input, and how the system should behave in particular situation. The functional requirements for a system should describe what the system do. These requirements depend on the type of software being developed, the expected users of the software and the general approach taken by the organization when writing requirements. Each high level functional requirement may involve several interactions or dialogues between the system and the outside world. There are many ways of expressing functional requirements for e.g., natural language, a structured or formatted language with no rigorous syntax and formal specification language with proper syntax.


Non-Functional Requirements

Non-functional requirements are the requirements that are not directly concern with the specific function of the system. They define system properties and constraints like readability, response time and storage requirements. It describes how the system will do it. The process of specifying non-functional requirements requires the knowledge of functionality of system, as well as the knowledge of context within which the system will operate.

Domain Requirements

Domain requirements are the requirements which are characteristic of a particular category or domain of projects. The basic functions that a system of a specific domain must necessarily exhibit come under this category. For instance, in an academic software that maintains records of a school or college, the functionality of being able to access the list of faculty and list of students of each grade is a domain requirement. These requirements are therefore identified from that domain model and are not user specific.

5. What are the five desirable characteristics of a good software requirements specification (SRS) document?

 **hide solution**
asked in 2068(II)


Solution

A **software requirements specification (SRS)** is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development.

Following are the five desirable characteristics of SRS:

- 1. Correctness:** User review is used to ensure the correctness of requirements stated in the SRS. SRS is said to be correct if it covers all the requirements that are actually expected from the system.
- 2. Completeness:** Completeness of SRS indicates every sense of completion including the numbering of all the pages, resolving the to be determined parts to as much extent as possible as well as covering all the functional and non-functional requirements properly.
- 3. Consistency:** Requirements in SRS are said to be consistent if there are no conflicts between any set of requirements. Examples of conflict include differences in terminologies used at separate places, logical conflicts like time period of report generation, etc.
- 4. Unambiguousness:** SRS is unambiguous when every fixed requirement has only one interpretation. This suggests that each element is uniquely interpreted. In case there is a method used with multiple definitions, the requirements report should determine the implications in the SRS so that it is clear and simple to understand.
- 5. Modifiability:** SRS should be made as modifiable as likely and should be capable of quickly obtain changes to the system to some extent. Modifications should be perfectly indexed and cross-referenced.

5. What do you mean by behavioral specification?

 **hide solution**
asked in 2069

Solution

The simple algebraic technique can be used to describe interfaces where the object state changing depending on the previous operation result. Where this condition holds we say it the behavior properties of system. The specification which is used to specify such type of system property is called behavioral specification.

The object state of the subsystem changes on the result of operations in the object or while interacting with other sub-system objects. The changes of the state of object is called behavioral characteristics of the system. The behavioral state of the formal specification is represented by model-based approach and is expressed as a state model.

Basic types of behavioral specification:

- 1. Abstract model specifications:** defines operations in terms of well-defined mathematical model.

2. **Algebraic specifications:** defines operations by a collection of equivalence relations.


3. **State transition specification:** defines operations in terms of states and transitions.

4. **Axiomatic specifications:** defines operations by logical assertions.

3. Differentiate between functional and non-functional requirements. What are the various types of functional and non-functional requirements that are placed on the system? Explain with example.

asked in Model Question

4. Differentiate between functional and non-functional requirements of software engineering requirements.

 **hide solution**
asked in 2075


Solution

Functional requirements are the statements of services the system should provide, how the system should react to particular input, and how the system should behave in particular situation. The functional requirements for a system should describe what the system do.

Non-functional requirements are the requirements that are not directly concern with the specific function of the system. They define system properties and constraints like readability, response time and storage requirements. It describes how the system will do it.

Functional Requirements	Non-functional Requirements
It defines external behavior of the system.	It defines the properties and constraints of the system.
Functional requirements are specified by user.	Non-functional requirements are specified by technical person, software developers, technical leader.
These requirements are mandatory.	These requirements may not be mandatory.
Helps to verify the functionality of software.	Helps to verify performance of software.
Describes what the product (i.e. software) does.	Describes how the product work.
It uses functional testing like system testing, integration testing etc.	It uses non-functional testing like performance, stress, security, usability, reliability testing etc.

5. What do you mean by software requirement? Explain the requirements engineering process with example.

 **hide solution**
asked in 2071(II)

Solution

The requirement for a system are the descriptions of what the system should do - the services that it provides and the constraints on its operation. Requirement may range from a high-level abstract statement of a services or of a system constraint to detailed mathematical specification.

Requirement engineering process:

Requirement engineering is the process of defining, documenting and maintaining the requirements in the engineering design process. The requirement engineering process consists of the following activities:

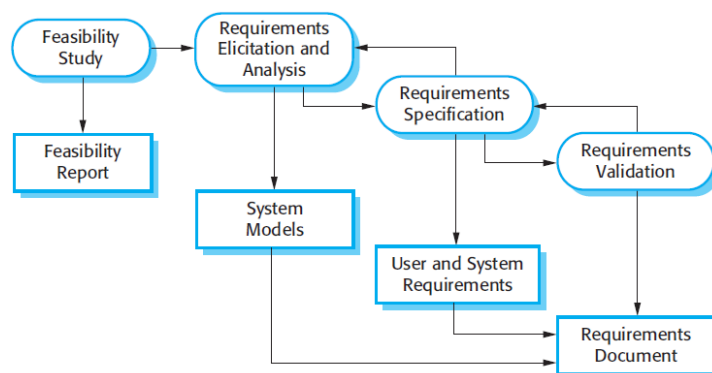


Fig: The requirements engineering process.


1. **Feasibility study:** An estimate is made of whether the identified can be achieved using the current software and hardware technologies, under the current budget, etc. The feasibility study should be cheap and quick; it should inform the decision of whether or not to go ahead with the project.

2. **Requirements elicitation and analysis:** This is the process of deriving the system requirements through observation of existing systems, discussions with stakeholders, etc. This may involve the development of one or more system models and prototypes that can help us understanding the system to be specified.

3. **Requirements specification:** It's the activity of writing down the information gathered during the elicitation and analysis activity into a document that defines a set of requirements. Two types of requirements may be included in this document; user and system requirements.

4. Requirements validation: It's the process of checking the requirements for realism, consistency and completeness. During this process, our goal is to discover errors in the requirements document. When errors are found, it must be modified to correct these problems.

5. What is the need of feasibility study ? Explain the various types of feasibility study with example.

 **hide solution**
asked in 2073

Solution

Feasibility study is so important stage software life cycle as after completion of feasibility study it gives a conclusion of whether to go ahead with proposed project as it is practically feasible or to stop proposed project here as it is not right/feasible to develop or to think/analyze about proposed project again.

Along with this Feasibility study helps in identifying risk factors involved in developing and deploying system and planning for risk analysis also narrows the business alternatives and enhance success rate analyzing different parameters associated with proposed project development.

Types of feasibility study:

1. Technical Feasibility: Technical feasibility helps in accessing the current resources as well as technology that is required for accomplishing the requirements of the user in the software within the allocated budget and time. Following tasks are performed in the technical feasibility:


- Helps in determining whether the technology used is stable.
- Is the technology being demanded available? Or can the features being requested are possible to embed with the current/ available technology.
- Analyze the technical capabilities and skills of the team members of software development.

2. Operational/ Behavioral Feasibility: Behavioral feasibility is studied in order to check, whether the human or employees in the business will use it or not. Operational feasibility relies on human resources and analyzes whether the software will operate after it is developed properly or not.

3. Economic Feasibility: Economic Feasibility helps in determining whether the required software has the potential to generate financial gains for an organization. This type of study involves the cost incurred on the team of the software development, cost of study involved in conducting a feasibility study, estimated cost of software and hardware.

4. Scheduling feasibility: It is the most important study that is used for measuring the success of the project. All the projects will fail, if not completed at specified time. In scheduling feasibility, an organization is able to estimate the time required for completing the project. The time that will be required for the development of the system, is estimated on the basis of the requirements. Once all these things are sorted, manpower is allocated their respective task and efforts are made to complete the task within the deadline.

6. Why do we need formal specification? Discuss behavioral specification in detail.

 **hide solution**
asked in 2071

Solution

A **formal software specification** is a statement expressed in a language whose vocabulary, syntax, and semantics are formally defined. It is a technique for unambiguous specification of software to be built. The specification languages cannot be based on natural language; it must be based on mathematics because natural language specifications are informal and usually contain ambiguities.

Formal methods are intended to systematize and introduce rigor into all the phases of software development. This helps us to avoid overlooking critical issues, provides a standard means to record various assumptions and decisions, and forms a basis for consistency among many related activities. By providing precise and unambiguous description mechanisms, formal methods facilitate the understanding required to coalesce the various phases of software development into a successful endeavor.

Behavioral specification

The simple algebraic technique can be used to describe interfaces where the object state changes depending on the previous operation result. Where this condition holds we say it the behavior properties of system. The specification which is used to specify such type of system property is called behavioral specification.

The object state of the subsystem changes on the result of operations in the object or while interacting with other sub-system objects. The changes of the state of object is called behavioral characteristics of the system. The behavioral state of the formal specification is represented by model-based approach and is expressed as a state model.

Basic types of behavioral specification:


1. Abstract model specifications: defines operations in terms of well-defined mathematical model.

2. Algebraic specifications: defines operations by a collection of equivalence relations.

3. State transition specification: defines operations in terms of states and transitions.

4. **Axiomatic specifications:** defines operations by logical assertions.

6. What are the different types of requirement elicitation techniques? Explain in brief.

 **hide solution**
asked in 2073


Solution

Requirement elicitation is the process of collecting the requirements of a system or requirement gathering from user, customers and stakeholders by conducting meetings, interviews, questionnaires, brainstorming sessions, prototyping etc.

Requirement elicitation techniques are:

- 1. Document analysis:** Document analysis is one of the most helpful elicitation techniques in understanding the current process. Documents like user manuals, software vendor manuals, process documents about the current system can provide the inputs for the new system requirements.
- 2. Observation:** This elicitation technique helps in collecting requirements by observing users or stakeholders. This can provide information about the existing process, inputs and outputs. There are two kinds of observations — active and passive. In **active observation**, the business analyst directly observes the users or stakeholders, whereas in **passive observation** the business analyst observes the subject matter experts.
- 3. Interview:** An interview is a systematic approach to elicit information from a person or group of people. In this case, the business analyst acts as an interviewer. An interview provides an opportunity to explore and/or clarify requirements in more detail. Without knowing the expectations and goals of the stakeholders it is difficult to fulfil requirements.
- 4. Brainstorming:** Brainstorming is an efficient way to define their requirements. Users can come up with very innovative ideas or requirements. This can help gather ideas and creative solutions from stakeholders in a short time. Users or stakeholders can come up with ideas that they have seen or experienced elsewhere. These ideas can be reviewed and the relevant ones can then be included in the system requirements.
- 5. Questionnaires:** Questionnaires are useful when there is a lot of information to be gathered from a larger group of stakeholders. This enables the business team to gather requirements from stakeholders remotely. The design of the questionnaire is very important, since it can influence the answers that people provide.
- 6. Prototyping:** Prototyping is building user interface without adding detail functionality for user to interpret the features of intended software product. It helps giving better idea of requirements. If there is no software installed at client's end for developer's reference and the client is not aware of its own requirements, the developer creates a prototype based on initially mentioned requirements. The prototype is shown to the client and the feedback is noted. The client feedback serves as an input for requirement gathering.

6. Why formal specification is important for software development? Explain.

 **hide solution**
asked in 2074

Solution

A **formal software specification** is a statement expressed in a language whose vocabulary, syntax, and semantics are formally defined. It is a technique for unambiguous specification of software to be build. The specification languages cannot be based on natural language; it must be based on mathematics because natural language specification are informal and usually contain ambiguities.

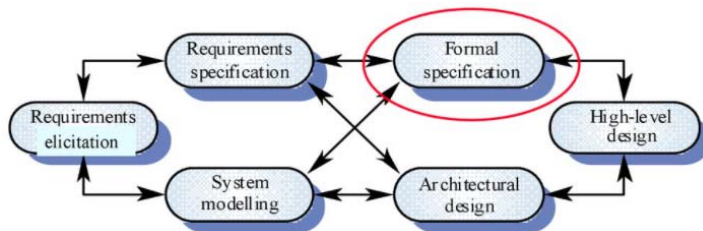



Fig: Formal specification in software process

The system requirements and system design are expressed in details and carefully analyzed and checked before implementation begins. A formal specification of software is developed after the system requirement have been specified but before the detailed system design. The main benefit of formal specification is its ability to uncover problem and ambiguities in the requirements specification. It forces to system analysis to remove errors and inconsistencies in the requirement specification.

Formal methods are intended to systematize and introduce rigor into all the phases of software development. This helps us to avoid overlooking critical issues, provides a standard means to record various assumptions and decisions, and forms a basis for consistency among many related activities. By providing precise and unambiguous description mechanisms, formal methods facilitate the understanding required to coalesce the various phases of software development into a successful endeavor.

Two fundamental techniques for formal specification are:

- 1. Algebraic Approach:** In algebraic approach, system is described in terms operation and their relationship. It consists of two parts: signature, which determines syntax of operation and an equation, which defines the semantics of operations.

 **hide solution**
asked in 2076

A **formal software specification** is a statement expressed in a language whose vocabulary, syntax, and semantics are formally defined. It is a technique for unambiguous specification of software to be build. The specification languages cannot be based on natural language; it must be based on mathematics because natural language specification are informal and usually contain ambiguities.

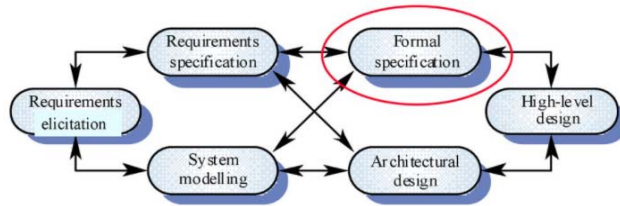


Fig: Formal specification in software process


Large systems are decomposed into subsystems with well-defined interfaces between these subsystems. Specification of subsystem interfaces allows independent development of the different subsystem. Subsystem make use of other subsystem, so an essential part of specification is to define subsystem. Once the interface are agreed and defined, the subsystem can then be designed and implemented independently. Subsystem interface are often defined as a set of object or component.

Three types of interface may have to be defined:

1. **Procedural interface:** Used for calling the existing programs by the new programs.
2. **Data Structure:** Provide data passing from one sub-system to another.
3. **Data representation:** Ordering of bits to match with the existing system.

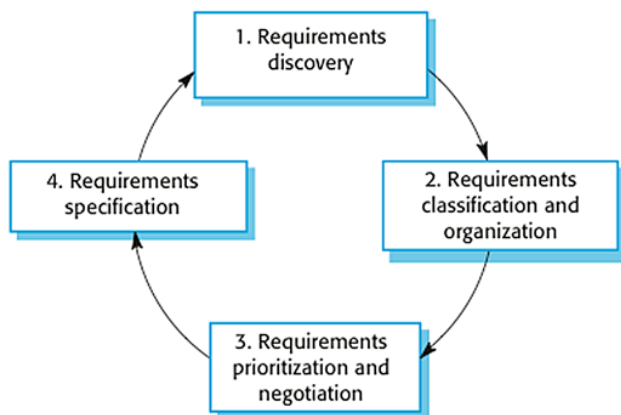
Formal notations are an effective technique for interface specification.

6. Why elicitation and analysis is a difficult process in requirement engineering process? Explain.

 **hide solution**
asked in 2068

Requirements elicitation and analysis is a process of interacting with customers and end-users to find out about the domain requirements, what services the system should provide, and the other constraints.

The requirements elicitation and analysis has 4 main process:



The users, customers or other stakeholders involved in the software development do not know all the features and behavior they need to have in the software at the time of requirements gathering. End-users at first encounter are most likely to give unrealistic requirements or requirements that are filled with ambiguities. The term elicitation is a pointer to the fact that good requirements cannot be obtained simply by asking the user, customer or stakeholders what they want. Requirements elicitation therefore calls for multiple interviews, questionnaires, user observations, brainstorming sessions and lots of prototypes. This is why requirements elicitation is difficult.

Solution

A **formal software specification** is a statement expressed in a language whose vocabulary, syntax, and semantics are formally defined. It is a technique for unambiguous specification of software to be build. The specification languages cannot be based on natural language; it must be based on mathematics because natural language specification are informal and usually contain ambiguities.

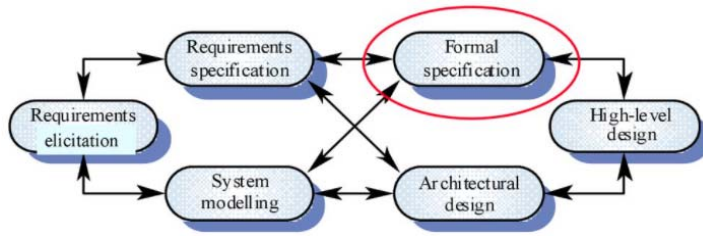


Fig: Formal specification in software process

The system requirements and system design are expressed in details and carefully analyzed and checked before implementation begins. A formal specification of software is developed after the system requirement have been specified but before the detailed system design.

The main benefit of formal specification is its ability to uncover problem and ambiguities in the requirements specification. It forces to system analysis to remove errors and inconsistencies in the requirement specification.

Two fundamental techniques for formal specification are:

- 1. Algebraic Approach:** In algebraic approach, system is described in terms operation and their relationship. It consists of two parts: signature, which determines syntax of operation and an equation, which defines the semantics of operations.
- 2. Model-Based Approach:** In model based approach, the abstract model of system is built using mathematical construct such as set theory, function theory and logic. It specifies the operations performed on abstract model.

Solution

A **software requirements specification (SRS)** is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consist of all necessary requirements required for project development.

Following are the characteristics of SRS:

- 1. Correctness:** User review is used to ensure the correctness of requirements stated in the SRS. SRS is said to be correct if it covers all the requirements that are actually expected from the system.
- 2. Completeness:** Completeness of SRS indicates every sense of completion including the numbering of all the pages, resolving the to be determined parts to as much extent as possible as well as covering all the functional and non-functional requirements properly.
- 3. Consistency:** Requirements in SRS are said to be consistent if there are no conflicts between any set of requirements. Examples of conflict include differences in terminologies used at separate places, logical conflicts like time period of report generation, etc.
- 4. Unambiguousness:** SRS is unambiguous when every fixed requirement has only one interpretation. This suggests that each element is uniquely interpreted. In case there is a method used with multiple definitions, the requirements report should determine the implications in the SRS so that it is clear and simple to understand.
- 5. Modifiability:** SRS should be made as modifiable as likely and should be capable of quickly obtain changes to the system to some extent. Modifications should be perfectly indexed and cross-referenced.
- 6. Verifiability:** SRS is correct when the specified requirements can be verified with a cost-effective system to check whether the final software meets those requirements. The requirements are verified with the help of reviews.
- 7. Traceability:** The SRS is traceable if the origin of each of the requirements is clear and if it facilitates the referencing of each condition in future development or enhancement documentation.

Solution


Functional Requirements

Functional requirements are the statements of services the system should provide, how the system should react to particular input, and how the system should behave in particular situation. The functional requirements for a system should describe what the system do. These requirements depend on the type of software being developed, the expected users of the software and the general approach taken by the organization when writing requirements. Each high level functional requirement may involve several interactions or dialogues between the system and the outside world. There are many ways of expressing functional requirements for e.g., natural language, a structured or formatted language with no rigorous syntax and formal specification language with proper syntax.

Non-Functional Requirements

Non-functional requirements are the requirements that are not directly concern with the specific function of the system. They define system properties and constraints like readability, response time and storage requirements. It describes how the system will do it. The process of specifying non-functional requirements requires the knowledge of functionality of system, as well as the knowledge of context within which the system will operate.

8. What do you mean by formal specification? Explain.

 **hide solution**
asked in 2068

Solution

A **formal software specification** is a statement expressed in a language whose vocabulary, syntax, and semantics are formally defined. It is a technique for unambiguous specification of software to be build. The specification languages cannot be based on natural language; it must be based on mathematics because natural language specification are informal and usually contain ambiguities.

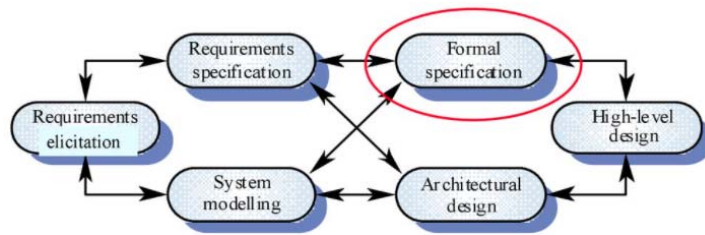


Fig: Formal specification in software process


The system requirements and system design are expressed in details and carefully analyzed and checked before implementation begins. A formal specification of software is developed after the system requirement have been specified but before the detailed system design.

The main benefit of formal specification is its ability to uncover problem and ambiguities in the requirements specification. It forces to system analysis to remove errors and inconsistencies in the requirement specification.

Two fundamental techniques for formal specification are:

- 1. Algebraic Approach:** In algebraic approach, system is described in terms operation and their relationship. It consists of two parts: signature, which determines syntax of operation and an equation, which defines the semantics of operations.
- 2. Model-Based Approach:** In model based approach, the abstract model of system is built using mathematical construct such as set theory, function theory and logic. It specifies the operations performed on abstract model.

10. Differentiate between interface specification and behavioral specification.

 **hide solution**
asked in 2068(II)

Solution

Interface Specification

Large systems are decomposed into subsystems with well-defined interfaces between these subsystems. Specification of subsystem interfaces allows independent development of the different subsystem. Subsystem make use of other subsystem, so an essential part of specification is to define subsystem. Once the interface are agreed and defined, the subsystem can then be designed and implemented independently. Subsystem interface are often defined as a set of object or component.

Three types of interface may have to be defined:

- 1. Procedural interface:** Used for calling the existing programs by the new programs.
- 2. Data Structure:** Provide data passing from one sub-system to another.
- 3. Data representation:** Ordering of bits to match with the existing system.

Formal notations are an effective technique for interface specification.

Behavioral specification

The simple algebraic technique can be used to describe interfaces where the object state changing depending on the previous operation result. Where this condition holds we say it the behavior properties of system. The specification which is used to specify such type of system property is called behavioral specification.

The object state of the subsystem changes on the result of operations in the object or while interacting with other sub-system objects. The changes of the state of object is called behavioral characteristics of the system. The behavioral state of the formal specification is represented by model-based approach and is expressed as a state model.

Basic types of behavioral specification:

1. Abstract model specifications: defines operations in terms of well-defined mathematical model.

2. Algebraic specifications: defines operations by a collection of equivalence relations.

3. State transition specification: defines operations in terms of states and transitions.

4. Axiomatic specifications: defines operations by logical assertions

Unit: 5 System Modeling

4 questions

Unit: 6 Architectural Design

12 questions

Unit: 7 Design and Implementation

10 questions

Unit: 8 Software Testing

26 questions

Unit: 9 Software Evolution

4 questions

Unit: 10 Software Management

14 questions

Suggest Us

Please give us feedback and suggestions to improve collegenote.

collegenoteofficial@gmail.com

Help

[About Us](#)

[Term & conditions](#)

[Privacy Policy](#)

Links and Resourses

[CSIT](#)

[BIT](#)

[BCA](#)

[Jobs and Internships](#)

[Blog](#)

[Contact](#)

Find us @

