

Arbind Mandal Kist
Unit 2 Software processes

1. What is software Evolution? 2017 kist

Software evolution is the term used in software engineering (specifically software maintenance) to refer to the process of developing software initially, then repeatedly updating it for various reasons.

2. List the phases of rational unified process.

RUP is a software development process from Rational, a division of IBM. It divides the development process into four distinct phases that each involve business modeling, analysis and design, implementation, testing and deployment. The four phases are:

- Inception – the idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.*
- Elaboration – the project's architecture and required resources are further evaluated. Developers consider possible applications of the software and costs associated with the development.*
- Construction – the project is developed and completed. The software is designed, written and tested.*
- Transition – the software is released to the public. Final adjustments or updates are made based on feedback from end users.*

3. What do you mean by prototyping?

Prototyping refers to an initial stage of a software release in which developmental evolution and product fixes may occur before a bigger release is initiated. These kinds of activities can also sometimes be called a beta phase or beta testing, where an initial project gets evaluated by a smaller class of users before full development.

4. .What is software prototyping?

Software prototyping is the activity of creating prototyping of software applications i.e. incomplete versions of software program being developed. It is an activity that can occur in software development and is comparable to prototyping as known from other fields, such as mechanical engineering or manufacturing.

5. What is prototype?

A prototype is an original model, form or an instance that serves as a basis for other processes. In software technology, the term prototype is a working example through which a new model or a new version of an existing product can be derived.

6. What is the importance of transition phase of RUP? (2017)

The importance of transition phase of Rup is: during transition, the project team focuses on correcting defects and modifying the system to correct previously unidentified problems.

8. Why spiral model is known as meta model.

Spiral Model is called meta model because it is composed of several other models. For example a single loop spiral actually represents the Waterfall Model. The spiral model uses prototyping approach before embarking on the actual product development effort. This enables the developer to understand and resolve the risks at each evolutionary level. The spiral model uses prototyping as a risk reduction mechanism and also retains the systematic step approach of waterfall model.

7. Define incremental development.

Incremental Development deals with design and development phase to incorporate changing requirements. Requirements of the customers are prioritized and highest priority requirements are developed in initial increments. Customer involvement in development may be require to prioritize their needs.

9. Define feasibility study/report and list its types.

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. Some of its types are:

- a. Technical feasibility*
- b. Operational feasibility*
- c. Economic feasibility*

Unit 3 Agile Software development

11. What is Scrum & How many phases are there in Scrum? 2017 Asian,Kist

It is the most popular agile framework, which concentrates particularly on how to manage tasks within a team-based development environment. Scrum uses iterative and incremental development model, with shorter duration of iterations.

The phases in Scrum are as follows:

- 1. Product Backlog Creation*
- 2. Sprint Planning and Sprint Backlog Creation*
- 3. Working on the Sprint. Scrum Meetings*
- 4. Testing and Product Demonstration*
- 5. Retrospective and Next Sprint Planning*

12. What is extreme programming & its principles .2018 1st terminal Asian, 2017 Hetauda

Extreme programming is mostly use agile method in which best practices of agile approach such as pair programming, test driven development, user involvement etc are used to extreme level.

The fundamental principles of Extreme Programming are –

- Rapid feedback /Assume simplicity /Incremental change/Embracing change /*
- Quality work*

13. Define the term refactoring g. 2017 Hetauda

Refactoring is the process of altering an application's source code without changing its external behavior. The purpose of code refactoring is to improve some of the nonfunctional properties of the code, such as readability, complexity, maintainability and extensibility. Refactoring can extend the life of source code, preventing it from becoming legacy code. The refactoring process makes future enhancements to such code a more pleasant experience. Refactoring is also known as reengineering.

13. Define pair programming(1st terminal Nagarjuna)

Pair programming is an Agile technique originating from extreme programming in which two developers team together and work on one computer. The two people work together to design, code and test user stories.

Arbind Mandal Kist

14. Difference between plan driven & a gile driven .(1st terminal Nagarjuna)

15. Define unit testing.(1st term kist)

Unit testing is a [software](#) development process in which the smallest testable parts of an [application](#), called units, are individually and independently scrutinized for proper operation. Unit testing can be done manually but is often [automated](#). Unit testing involves only those characteristics that are vital to the performance of the unit under test. This encourages developers to modify the [source code](#) without immediate concerns about how such changes might affect the functioning of other units or the program as a whole.

16. Define sprints.(1st term kist.)

A sprint (or iteration) is the basic unit of development in Scrum. The sprint is a [timeboxed](#) effort; that is, it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month, with two weeks being the most common. Each sprint starts with a sprint planning event that aims to define a sprint backlog, identify the work for the sprint, and make an estimated forecast for the sprint goal. Each sprint ends with a sprint review and sprint retrospective, that reviews progress to show to stakeholders and identify lessons and improvements for the next sprints

Unit 4 Requirements engineering

19. What is requirement engineering? (2016, 1st term kist)

Requirements engineering is a process of gathering and defining of what the services should be provided by the system. It focuses on assessing if the system is useful to the business (feasibility study), discovering requirements (elicitation and analysis), converting these requirements into some standard format (specification), and checking that the requirements define the system that the customer wants (validation).

17. What is requirement validation? List the types of check conducted during requirement validation(1st terminal Nagarjuna)

It's a process of ensuring the specified requirements meet the customer needs. It's concerned with finding problems with the requirements

- 1. **Validity** checks: The functions proposed by stakeholders should be aligned with what the system needs to perform. You may find later that there are additional or different functions are required instead.*
- 2. **Consistency** checks: Requirements in the document shouldn't conflict or different description of the same function*
- 3. **Completeness** checks: The document should include all the requirements and constraints.*
- 4. **Realism** checks: Ensure the requirements can actually be implemented using the knowledge of existing technology, the budget, schedule, etc.*
- 5. **Verifiability**: Requirements should be written so that they can be tested. This means you should be able to write a set of tests that demonstrate that the system meets the specified requirements.*

21. Define user requirement & system requirement(1st term kist)

User requirements are functional requirements. They deal with functionality that is visible and important to users that a system has to deliver to satisfy the business objectives that the system is designed to fulfill. System requirements are non-functional requirements. They deal with the characteristics that the system must meet to provide a sufficient level of reliability, usability, maintainability, availability..

22. Define enduring requirements with example. 2014

Enduring requirements *These are relatively stable requirements that derive from the core activity of the organization and which relate directly to the domain of the system. For example, in a hospital there will always be requirements concerned with patients, doctors, nurses, treatments, etc. These requirements may be derived from domain models that show the entities and relations which characterise an application domain.*

Unit 5 System modeling

23. Define context model? 2017 cab

A **context model** (or context modeling) defines how context data are structured and maintained (It plays a key role in supporting efficient context management). It aims to produce a formal or semi-formal description of the context information that is present in a context-aware system. In other words, the context is the surrounding element for the system, and a model provides the mathematical interface and a behavioral description of the surrounding environment. It is used to represent the reusable context information of the components .

24. What is use case modeling?

A use case can be taken as a simple scenario that describes what a user expects from a system. Each use case represents a discrete task that involves external interaction with a system. In its simplest form, a use case is shown as an ellipse with the actors involved in the use case represented as stick figures.

25. Explain generalization . 2017 cab

Generalization is an everyday technique that we use to manage complexity. Rather than learn the detailed characteristics of every entity that we experience, we place these entities in more general classes (animals, cars, houses, etc.) and learn the characteristics of these classes. This allows us to infer that different members of these classes have some common characteristics (e.g., squirrels and rats are rodents). We can make general statements that apply to all class members.

26. What do you mean by system modeling?(2015)

System modelling is the process of developing abstract models of a system, with each model presenting a different view or perspective of that system. System modelling has generally come to mean representing the system using some kind of graphical notation, which is now almost always based on notations in the Unified Modelling Language (UML). However, it is also possible to develop formal (mathematical) models of a system, usually as a detailed system specification

28. Define sequence diagram. 1st term Asian

sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

29. What is modular decomposition?(2015,2014,2013)

structural level where sub-systems are decomposed into modules .Two modular decomposition models covered An object model where the system is decomposed into interacting objects A data-flow model where the system is decomposed in.

30. Which model in system modelling depicts the static nature of the system ? 2017 kist

Structural models show the organization and architecture of a system. These are used to define the static structure of classes in a system and their associations

Arbind Mandal Kist

Unit 6 Architectural design

31. What is pipe and filter architecture?

This is a model of the run-time organization of a system where functional transformations process their inputs and produce outputs. Data flows from one to another and is transformed as it moves through the sequence. Input data flows through these transforms until converted to output. The transformations may execute sequentially or in parallel. The data can be processed by each transform item by item or in a single batch.

32. Define design pattern.(2017)

*A **software design pattern** is a general, reusable solution to a commonly occurring problem within a given context in software design. It is not a finished design that can be transformed directly into source or machine code. It is a description or template for how to solve a problem that can be used in many different situations. Design patterns are formalized best practices that the programmer can use to solve common problems when designing an application or system.*

33. Define repository model. (2017,2013)

Repository pattern describes how a set of interacting components can share data. The majority of systems that use large amounts of data are organized around a shared database or repository. This model is therefore suited to applications in which data is generated by one component and used by another. All data in a system is managed in a central repository that is accessible to all system components. Components do not interact directly, only through the repository.

Unit 8 Software testing

34. What is the importance of verification and validation? 2017 cab

The ultimate goal of verification and validation processes is to establish confidence that the software system is 'fit for purpose'. This means that the system must be good enough for its intended use. The level of required confidence depends on the system's purpose, the expectations of the system users, and the current marketing environment for the system:

35. What is test driven development. 2017 cab

Test-driven development (TDD) is an approach to program development in which you interleave testing and code development. Essentially, you develop the code incrementally, along with a test for that increment. You don't move on to the next increment until the code that you have developed passes its test.

36. What is alpha testing?

An alpha test is a preliminary software field test carried out by a team of users in order to find bugs that were not found previously through other tests. The main purpose of alpha testing is to refine the software product by finding (and fixing) the bugs that were not discovered through previous tests.

37. What is regression testing? 2017 cab

Regression testing is the process of testing changes to computer programs to make sure that the older programming still works with the new changes. Regression testing is a normal part of the program development process and, in larger companies, is done by code testing specialists. Test department coders develop code test scenarios and exercises that will test new units of code after they have been written.

38. What is release testing?

Release testing is the process of testing a particular release of a system that is intended for use outside of the development team. Normally, the system release is for customers and users. In a complex project, however, the release could be for other teams that are developing related systems. The primary goal of the release testing process is to convince the supplier of the system that it is good enough for use.

39. Write the benefits of involving end-users in release testing.(2017)

- End users understand exactly what the system needs to do (for them)*
- End users' acceptance is more likely if they're involved during earlier stages of testing.*
- End users can identify bugs before they get too costly to fix*
- End users offer a broader perspective.*

Unit 9 Software evolution

40. What is legacy system?

It refers to outdated computer systems, programming languages or application software that are used instead of available upgraded versions. Legacy systems also may be associated with terminology or processes that are no longer applicable to current contexts or content, thus creating confusion. In theory, it would be great to be able to have immediate access to use the most advanced technology. But in reality, most organizations have legacy systems - to some extent. A legacy system may be problematic, due to compatibility issues, obsolescence or lack of security support.

41. Define software evolution.(2016 ,2017 Hetauda)

The stage in a software system's life cycle where it is in operational use and is evolving as new requirements are proposed and implemented in the system. Software evolution is the term used to define the process that starts by the development of the software and by any step that incrementally updates the software

42. What is System engineering? 2017 Cab

It focuses on how to design and manage complex systems over their [life](#) cycles. At its core, systems engineering utilizes systems thinking principles to organize this body of knowledge. Issues such as requirements engineering, reliability, logistics, coordination of different teams, testing and evaluation, maintainability and many other disciplines necessary for successful system development, design, implementation, and ultimate decommission become more difficult when dealing with large or complex projects. Systems engineering deals with work-processes, optimization methods, and risk management tools in such projects.

43. List the different types of software maintenance.(2016, 2017 kist)

- *Fault repairs:* Coding errors are usually relatively cheap to correct; design errors are more expensive as they may involve rewriting several program components..
- *Environmental adaptation:* This type of maintenance is required when some aspect of the system's environment such as the hardware, the platform operating system, or other support software changes.
- *Functionality addition:* This type of maintenance is necessary when the system requirements change in response to organizational or business change.

Unit 10 Sociotechnical systems

45. Define socio-technical system.(2017) 2017 kist

Socio-technical systems design (STSD) methods are an approach to design that consider human, social and organisational factors, as well as technical factors in the design of organisational systems. Systems that include technical systems but also operational processes and people who use and interact with the technical system. Socio-technical systems are governed by organisational policies and rules.

47. What is system procurement? 2014

The initial phase of systems engineering is system procurement (sometimes called system acquisition). At this stage, decisions are made on the scope of a system that is to be purchased, system budgets and timescales, and the high-level system requirements. Using this information, further decisions are then made on whether to procure a system, the type of system required, and the supplier or suppliers of the system.

48. What is system evolution? 2017 Asian

Hence evolution, where the system changes to accommodate environmental change, is a process that runs alongside normal system operational processes. System evolution involves re-entering the development process to make changes and extensions to the system's hardware, software, and operational processes.

49. What is complex system?

A characteristic of all complex systems is that the properties and behaviour of the system components are inextricably intermingled. The successful functioning of each system component depends on the functioning of other components. Thus, software can only operate if the processor is operational. The processor can only carry out computations if the software system defining these computations has been successfully installed.

50. Why formal method of software engineering? 2017 Asian

Formal methods in software engineering are mathematical techniques that are used in the design, implementation and testing of computer systems. Performing appropriate mathematical analysis can contribute to the reliability and robustness of a design

51. What is open source development? 2017 Asian

Open-source software development is the process by which open-source software, or similar software whose source code is publicly available, is developed. These are software products available with its source code under an open-source license to study, change, and improve its design. Examples of some popular open-source software products are Mozilla Firefox, Google Chromium, And the VLC media player.

Advantage

- Lesser hardware costs
- High-quality software
- No vendor lock-in
- integrated management
- Simple license management
- Lower software costs

57. What is CASE (Computer-Aided Software Engineering)? 2017 Hetauda

It is the domain of software tools used to design and implement applications. CASE tools are similar to and were partly inspired by computer-aided design (CAD) tools used for designing hardware products. CASE tools are used for developing high-quality, defect-free, and maintainable software.

58. What do you mean by test cases? 2017 Hetauda

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites.

59. What is software inspection? 2017 Hetauda

In software development, software or code inspection refers to peer review of any product by experts who try to unravel defects, if any, present in the system using a well defined method. In an inspection process, the objective is to identify defects, and the inspected parts are generally SRS (Software Requirements specifications) documents and test plans.

60. What is forward engineering? 2017 Hetauda

Forward engineering is a process of obtaining desired software from the specifications in hand which brought down by means of reverse engineering. It assumes that there was some software engineering already done in the past. Forward engineering is same as software engineering process with only one difference it is carried out always after reverse engineering.

61. What is Host-Target development concept?

Production software does not usually execute on the same computer as the software development environment. Rather, you develop it on one computer (the host system) and execute it on a separate computer (the target system).

62. Which of the mechanism that allows several objects in a class hierarchy to have different methods with the same name? 2017 kist

In polymorphism instances of each subclass will be free to respond to messages by calling their own version of the method.

63. Define exploratory development. (2015)

Exploratory development, to me, means trying out new development technologies or infrastructure solutions. It could also mean building new functionality from scratch, or as part of an existing system. Maybe you've been asked to look at how something might work, before getting official go ahead for a bigger piece of work. Incremental development is about progressively working your way through the development life cycle, usually meaning short development time periods, releasing more often etc.

64. How software quality is managed?

Software quality management (SQM) is a management process that aims to develop and manage the quality of software in such a way so as the best ensure the product meets the quality standards expected by the customer while also meeting any necessary regulatory and developer requirements, if any. Software quality managers require software to be tested before it is released to the market, and they do this using a cyclical process-based quality assessment in order to reveal and fix bugs before release.

65. Explain pricing to win technique for software cost estimation.

Price-to-Win identifies the correct balance of capability that can be delivered to the customer at the value the customer wants. PRICE-TO-WIN 8 STEP PROCESS

- *Step 1: Gather market intelligence: Marketing personnel identify competitors possible strategies and*
- *Step 2: Determine requirements & features*
- *Step 3: Sketch out Architecture Design*
- *Step 5. Determine viable alternatives*
- *Step 6: Cost Analysis & Estimation of Alternatives Step 7: Select best alternative Step 8: Establish price*

1. Define the term ethnography. (2017,2015,2014,1st term Asian,2017 cab)
2. Chapter 10 What are the organizational factors that may affect the design, requirements and operation of sociotechnical system? (2016)
3. Differentiate between safety and security.(2016 ,2017 Asian)
4. What is safety-critical system? **2017 Hetauda**
5. Differentiate between technology risk and people risk.2017 Asian
6. What is business risk?
7. What are the reliability metrics? 2017 Asian
8. Is it necessary to test the integration of modules if each of the modules are functioning properly? 2014
9. Is there any relation between quality and productivity?2013
- 10.How market environment affects software cost?2013, 2017 Asian

11. What do you mean by system decommissioning? 2013
12. What are two stimuli that affect behavior of system?(1st term kist)
13. What is quality planning?(2015)
14. List the activities of software measurement process.(2015)
15. Why is version management done?(2017)
16. Differentiate between module and system.(2015)
17. Differentiate between security and safety property of dependable software.
18. How market environment affects software cost?
19. What kind of decision will drive you towards procurement of a system? 2017 cab

20.

21.

22.