***SUBAIR CALEB IBUKUN***

***17/SCI01/078***

1ai)



1aii)

* Precision
* Systems are becoming increasingly dependent on software components so formal method is considered to accomplish this
* Measure of correctness
* Early defect detection
* Guarantees of correctness

1b) **Non-functional requirements** describe the general characteristics of a system. They are also known as quality attributes. It defines the quality attribute of a software system. It also represents a set of standards used to judge the specific operation of a system.

 WHILE

**Functional requirements** describe how a product must behave, what its features and functions. Functional software requirements help you to capture the intended behaviour of the system. This behaviour may be expressed as functions, services or tasks or which system is required to perform.

2a)

* Waterfall Model
* Prototype Methodology
* Agile Software Development Methodology
* Rapid Application Development
* Spiral Model

**Waterfall Model**: Considered as the traditional method of explaining the software development process in software engineering, waterfall model happens to clarify the process into a linear flow with a specified sequence to let the users understand that further level is made progressive on completion of the previous one. This methodology also emphasizes that going back to deal with the changes is not possible. Some advantages are; easy to understand and functional, simple enough to handle as model is rigid, saves significant amount of time, allows for easy testing and analysis etc. Disadvantages include; Only matches precise needs, not applicable for maintenance projects, no option to know possible outcome of a project, not excellent for long and ongoing projects.

**Prototype Methodology:** It is a specialized software development procedure that initiates developers towards making only the sample of the resolution to validate its functional essence to the customers and make essential changes before creating the authentic final solution. it tends to resolve a set of diversifying issues occurring with the waterfall method.

Advantages

Gives clear idea about the functional process of the software

Reduces the risk of failure in a software functionality

Disadvantages

Chances of extension in management cost

Excessive involvement of client can affect processing.

**Agile Software Development Methodology**: it is used for articulating a well-organized project management procedure allowing for recurrent alterations. It is a theoretical outline for undertaking several software engineering projects. it minimizes peril by creating software in short time boxes, known as iterations, which happen to last from one week to one month.

Advantages

Adaptive approach that responds to changes favorably

Allows for direct communication to maintain transparency

Disadvantages

Focuses on working with software and lacks documentation efficiency

Chances of getting off-track as outcome are not clear

**Rapid Application Development**: Aimed at providing quick results, rapid application development is meant to give excellent development processes with the assistance of other development approaches. It is created to take the maximum advantage from the development software. it is designed to augment the workability of the whole software development procedure for highlighting the participation of an active user.

Advantages.

Makes the entire development process effortless

Assists client in taking quick reviews

Disadvantages

Dependant on the team for performance

Requires extremely skilled personnel to handle complexities

**Spiral Model:** it is meant to reduce the early risks in the project. As per the process going by, the developers initiate on a smaller level and explore the included risks in it. Furthermore to this, the developers are intended towards crafting a plan for iterating of the spiral.The accomplishment of any Spiral Lifecycle model is based on consistent, observant, and conversant management of the project

Advantages

Risk factors are considerably reduced

Excellent for large and complex projects

Disadvantages

Costly model in software development

Failure in risk analysis phase may damage the whole project

2b)



3ai) Propositional logic is an analytical statement which is either true or false. It is a technique that represents the knowledge in logical & mathematical form.

first-order predicate logic is a collection of formal systems used in mathematics, and computer science. First-order logic uses quantified variables over non-logical objects and allows the use of sentences that contain variables, rather than propositions.

Propositional Logic converts a complete sentence into a symbol and makes it logical whereas in First-Order Predicate Logic relation of a particular sentence will be made that involves relations, constants, functions, and constants.

First-order predicate logic uses quantified variables while propositional logic, does not use quantifiers or relations

3aii)

-Negation: The negation of A is sometimes called the inverse of A. If A is a proposition, so is -A: -A is true when A is false, and -A is false when A is true

-The logical operation |, also called OR and logical disjunction, is an operation on two propositions (a binary operation) that results in another proposition: the proposition (B | C) is true if B is true or if C is true or if both A and B are true.

-The logical operation &, also called logical conjunction, combines two propositions to produce another. The proposition (A & C) is true if both A is true and C is true; it is false if either A is false or C is false (or both)

3b) “Every man is mortal. Smith is a man. Therefore, Smith is mortal”

P for the predicate "is a man."

Q to represent the predicate "is mortal" then Qx stands for "x is mortal."

We can then write the statement "If x is a man then x is mortal" as Px→Qx.

 "For all x, if x is a man then x is mortal"

Then the statement becomes;

 ∀x[Px→Qx].

4a) “There exists an object that is either a curious monkey or not a monkey at all”.

∀x ∃y P(x,y) ≠ ∃y ∀x P(X,y)

4b) The Z notation or specification: is a formal specification language used for describing and modelling computing systems. It is targeted at the clear specification of computer programs and computer-based systems in general. Z is based on the standard mathematical notation used in axiomatic set theory, lambda calculus, and first-order predicate logic. All expressions in Z notation are typed, thereby avoiding some of the paradoxes of naive set theory. Z contains a standardized catalogue (called the mathematical toolkit) of commonly used mathematical functions and predicates, defined using Z itself. Although Z notation (just like the APL language, long before it) uses many non-ASCII symbols, the specification includes suggestions for rendering the Z notation symbols in ASCII and in LaTeX. There are also Unicode encodings for all standard Z symbols

The Vienna Development Method (VDM) : is one of the longest-established formal methods for the development of computer-based systems. it has grown to include a group of techniques and tools based on a formal specification language—the VDM Specification Language (VDM-SL). It has an extended form, VDM++,[2] which supports the modelling of object-oriented and concurrent systems. Support for VDM includes commercial and academic tools for analysing models, including support for testing and proving properties of models and generating program code from validated VDM models.

The B method: is a method of software development based on B, a tool-supported formal method based on an abstract machine notation, used in the development of computer software. B is related to the Z notation (also originated by Abrial) and supports development of programming language code from specifications. B has been used in major safety-critical system applications in Europe (such as the automatic Paris Métro lines 14 and 1 and the Ariane 5 rocket). It has robust, commercially available tool support for specification, design, proof and code generation. ComparedCompared to Z, B is slightly more low-level and more focused on refinement to code rather than just formal specification — hence it is easier to correctly implement a specification written in B than one in Z.

5bi) A well-formed formula: (WFF) in propositional logic is a syntactically correct formula created

according to the syntactic rules of the underlying calculus. A well-formed formula is built up

from variables, constants, terms and logical connectives such as conjunction (and), disjunction

(or), implication (if … then …), equivalence (if and only if) and negation.

Quantifiers: are phrases that refer to given

quantities, such as "for some" or "for all" or "for every",

indicating how many objects have a certain property.

Terms: Terms are simply names for objects

6a) Implementation / Coding: On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer.

Testing: After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase.

Maintenance: Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance

6b) A requirement document defines in logical terms, how a system or project will accomplish the laid out requirements. It outlines the functionality of the system in detail by capturing the intended behaviour of the system, expressed as services, tasks or functions that the developers have agreed to provide.

7a)

7b) steps involved in writing a system requirement specification

Make an outline.

Define the purpose of your product.

Describe what you're building.

Detail the requirements.

Get it approved.