Software engineering lawas

Engineering

There may be several seasons for requirement deficiencies such as requirements may not be determined correctly or misinterpretation of requirements. Deficient requirements are Incorrect or Incomplete which results In failure of project. Example: Vultures (Software Company) released software. However, clients recognized that it do not work as per their requirements. After inspection, the result of failure of software was detected as requirements deficiencies.

Poem's First Law: " Errors are most frequent during the requirements and design activities and are the most expensive the later they are moved" Activities related to requirements and design and design are most heavily loaded with errors I. E. Errors are most obvious In requirement and design phases. The cost Incurred In removing errors in later parts of project will be much high. If the errors are removed in early parts, the cost will be less, as the time span of error increases the cost of removing the error also increases. In other words, the cost of removing errors is directly proportional to lifetime of error.

Example: Mentor Graphics Company released a software application with over 300 bugs in it. In order o remove the errors by expected deadline, the company hired more software engineers and thus the cost of project was Increased which was much higher than estimated cost. Poem's Second Law: " Prototyping (significantly) reduces requirement and design errors, especially for user interfaces" Prototyping is the best remedy to requirement and design errors. Prototyping is the toy implementation which helps to user to see how the project would appear after completion.

Therefore, requirement and design errors can be identified and removed. Example: Cloud Computing and web mining applications make use of prototyping model to reduce acquirement and design errors. Davis' Law: " The value of a model depends on the view taken, but none is best for all purposes". The importance of model relies on the way it is viewed as different model views include data, process, state transition, structure and behavior. However, single model cannot solve all the purposes as every model can answer different question in a different perspective.

For example, data view includes Data Flow Diagrams (DVD) and Entity Relationship Diagrams (RED). Doffs are used to describe flow of data among deferent entitles and Reds are used to scribe relationships among various entitles. In order to describe state transitions, state diagrams can be used. Any model can answer different questions that a project could face in real world. Example: Well-known company Ford motors created a real life size model of their upcoming vehicles using clay so as to observe the look and feel of actual project.

It also helps the engineers to collect the data about the aerodynamics of the car and the changes can be made by observing the model Queue: Your first task is to describe each software development methodology clearly and completely in your own words. You may use diagrams, examples or ML to help you do this ANSI: 1 . Waterfall Model: Waterfall Model is the basic software development methodology. It follows the systematic sequential approach starting from requirement analysis to maintenance by passing through other phases such as system design, implementation, testing, and deployment of system.

We have to follow sequentially, I. E. We cannot move to next phase until the requirements of first phase are met. Different phases of Waterfall model are: 1. Requirement Gathering and Analysis: The goal of this phase is to collect all elevate information regarding the product to be developed from the customer with a view to clearly understand the customer requirements. The customer requirements identified during requirements gathering and analysis activity are organized into a Software Requirement Specification (SIRS) document. 2.

System Design: The goal of design phase is to transform the requirements specified in the SIRS document into a structure that is suitable for implementation in some programming language. There are different design approaches such as structured design and object oriented design. Structured design includes Doffs. 3. Implementation: In implementation phase, the design is translated into source code. Each component of the design is implemented as a program module. The end-product of this phase is a set of program modules that have been individually tested. . Testing: In testing phase, first each module is tested in isolation from other modules, then debugging and documenting it. It is most efficient to debug the errors at identified at this stage. In testing phase, integration of different modules is undertaken once they have been coded and unit tested. During this phase, all the modules are integrated in a planned manner. 5. Deployment of System: This phase consists of all activities that make a system available for use. Deployment activities include release, install and activate.

Release includes all the operations to prepare a system for assembly and transfer to the customer site. Activation is the activity of starting up the

executable component of software. 6. Maintenance of System: Maintenance of system requires much more effort than the effort necessary to develop the product itself. 2. SPIRAL MODEL: Spiral Model is a software development process model that imbibes iterative nature of prototyping with controlled and systematic aspects of linear sequential model. Using Spiral Model, software is developed as series of incremental releases as there are several iterations.

During previous iterations, the incremental release may be a rough model or prototype and during later versions, the complete versions of software are developed. Spiral model is flexible in comparison to other models. Customer communication? Tasks are required to establish effective communication between developer and customer. Planning? Tasks are required to define sources, timeliness, and other project related information. Risk analysis? Tasks required to assess both technical and management risks. Engineering? Tasks are required to build one or more representations of the application. Construction and release? Tasks are required to construct, test, install, and provide user support (e. G. , documentation and training). Customer evaluation? Tasks are required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage. Sues 3: Using the first four laws of the text, show where these are either implemented or missing in each software development methodology (Total Two). If a law is missing, explain the consequences and suggest how the process might be improved.

ANSI: Waterfall Model in accordance with laws: Waterfall Model with Glass Law: Glass Law states that requirement deficiencies are major source of

project failures. So requirements should be clearly defined for successful projects. Glass Law is implemented in first phase of waterfall model which focus primarily on gathering and analysis of requirements. In waterfall model, we cannot move to next phase until the requirements of first phase are met. Waterfall Model with Poem's First Law: Poem's first law states that it is very expensive to remove errors in later stages.

Poem's Law is also implemented in waterfall model in requirement analysis phase which focuses to weed out the incompleteness and inconsistencies in gathered requirements. Waterfall Model with Poem's Second Law: Poem's Second Law is missing in Waterfall model as no prototype is build. As waterfall model includes gathering of requirements, creating design from gathered acquirement, converting the design into code, and then testing of code and maintenance of developed system. Poem's Second Law is missing, therefore various consequences are: I. It will result in requirement and design errors. It. There will be wastage of time. Ii. There will be misuse of resources. 'v. It may lead to failure of project. V. It may increase the cost of project. Actions to improve the process: I. Requirement elicitation and gathering must be done properly. It. Requirements gathered must be revised. Waterfall Model with Davis' Law: Davis' Law is implemented in second phase of tearful model I. E. Design Phase. Design Phase describe various models such as Doffs, Reds etc. To be used in different situations. Spiral Model in accordance with laws: Spiral Model with Glass Law: Glass Law is implemented in spiral model as spiral model is iterative in nature.

Therefore, requirements are revised in every phase. However, undiscovered errors may lead to project failures. Spiral Model with Poem's law is

implemented, as verification, validation and test plan are carried out there to cover errors in requirements and design phase. Spiral Model with Poem's Second Law: Poem's Second Law is implemented in spiral model as in every phase a prototype is built up and it reduces requirement and design errors. Spiral Model with Davis' Law: Davis' law is implemented in Spiral Model as different views can be used in prototyping in various phases.

Sues 4: For each software development methodology, give an example of a project which it would be well suited for and one which it would be inappropriate for (Total Two projects for each software development methodology). Justify your answers. ANSI: Example of Project in which Waterfall Model is used: While making smaller software such as Library Management, Waterfall model can be used. In library management project, first of all the requirements of library are analyses I. E. What is expected from library software in terms of function and behavior.

After analyzing requirements, the requirements are translated into design (I. E. In form of algorithm etc.). Then algorithm is translated into code. Coding can be in any programming language. After coding, the library software can be tested to check whether the requirements have been met and it is error free. Then, software is delivered to customer and makes the software working at customer's site. Example of project where Waterfall model is inappropriate: Waterfall model is not suitable for large real life projects as it works sequentially; we cannot move further after the first phase is over.

Waterfall model is not appropriate for aircraft system. Because it is very complex system and need may arise at any phase to make changes in the

products of previous phases. Example of Project in which Spiral Model is used: Spiral Model is used in development of TRW Software Productivity System. Three rounds (Round O, Round 1 and Round 2) were conducted to develop TRW Software Productivity System. Round O was able to answer basic feasibility questions and eliminate basic classes of candidate solutions. In round 1, the level of investment was greater, objectives, constraints and risks identified were more specific.

The risk-resolution activities were also more extensive. In round 2, there was top level requirement specification I. E. SERE tools were used for requirement specification and analysis. Spiral Model is used in development of Microsoft Operating System and Compilers. Example of Project in which Spiral Model in inappropriate: The usage of Spiral Model is inappropriate in projects that deal tit mathematical computations because in such projects the objectives are clear and requirements can be easily established. It will be time consuming and costly to use spiral model for such projects.

Sues 5: Describe and illustrate this law with examples. This law was published in 1965; is this law still valid today? Justify your answer with examples. NAS 5: Moor's Law is the observation that the history of computing hardware, the number of transistors that can be placed inexpensively on an integrated chips doubles approximately every 18 months that is the period for a doubling in chip performance. The capabilities of many digital electronic devices are strongly linked to Moor's law: processing speed, memory capacity, sensors and even the number and size of pixels in digital cameras.

All of these are improving at (roughly) exponential rates as well. The graph shows the rapid growth in number of transistors produced from 1980 to 2010. This growth is still increasing and therefore this law is still valid today as this law is applicable not merely to processors but to all the electronic devices, chips, transistors etc. Example: In January 1995, the Digital Alpha 21164 microprocessor had 9. Million transistors. Six years later, a state of the art microprocessor contained more than 40 million transistors.

It is theorists that with further miniaturization, by 201 5 these processors should contain more than 1 5 billion transistors, and by 2020 will be in molecular scale production, where each molecule can be individually positioned. Another example of Moor's La; the RAM is getting more powerful and cheaper day by day. As the number of cores in a chip goes up, its ability to process data begins to exceed bus technology ability to deliver it. But the speed is governed by Moor's Law.