Nickname: mike6606
Age: 8 years 10 months
Fans: 0
Followers: 5
+ Plus follow

< March 2023 >

day	One	Two	Three	Four	Five	Six
26	27	28	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Search

Look for it

Google search

Frequently used links

My essays
My comment
My involvement
Latest comments
My labels

Essay archives

January 2021 (1)
January 2018 (3)
January 2017 (8)

Read the leaderboard

1. Software Engineering Practitioners' Research Methods Chapter 13 Answers (1578)
2. Software Engineering Practitioners' Research Methods Chapter 1471 Answers (<>)
3. Software Engineering Research Methods for Practitioners Chapter 1290 Answers (<>)
4. Software Engineering Practitioners' Research Methods Chapter 34 Answers (1282)
5. Software Engineering Practitioners' Research Methods Chapter 1264 Answers (<>)

Recommended leaderboards

1. Software Engineering Practitioners' Research Methods Chapter 1 Answers (<>)

Software Engineering Practitioners' Research Methods Chapter 4 Answers

Problem:

Provide three examples of software projects that would be amenable to the waterfall model. Be specific.

Answer:
Waterfall model is a sequential approach to software development that begins with customer specification requirements and progresses through planning, modeling, construction and deployment

The waterfall model is appropriate for projects with the following characteristics:

(1) The problem is well understood (requirements are well-defined);

(2) The delivery date is realistic;

(3) it's unlikely that major changes in requirements will be requested as the project proceedsCommunication project initiation, requirements gathering

Planning Estimating, scheduling, tracking

Modeling Analysis and design

Construction code and test

Deployment delivery, support and feedback.

Example software project – (1) transaction Maintenance system (TMS)

(2) Us Department of defense

(3) NASA(1) Transaction Maintain once system

Phase 1. Preliminary investigation: Aim of this phase is not develop system but to

investigate the problem.

Phase 2. Requirement analysis: This phase is concerned abort the collection of

requirements. Out put of this phase is SRS document.

Phase 3. System design: (1) Algorithm

(2) Data structure

(3) software architecture

(4) Interface design

Phase 4. coding

Phase 5. Integration & Testing

Phase 6. Implementation & maintain once. In similar lines, us department of defense and NASA uses the waterfall model in an extensive way. As waterfall model is a sequential approach and many increments can be applied after the evolution of software in this model, it is well suited for large and long term projects.

Along with these software projects, many government projects follow waterfall model.

Problem:
Provide three examples of software projects that would be amenable to the prototyping model. Be specific.

Answer:
Software applications that are relatively easy to prototype almost always involve human-machine interaction.

When the customer has a legitimate need but is clueless about the details then develop a prototype as a first step. A customer defines a set of general objectives. For these

applications that are amenable to prototyping are certain classes of mathematical algorithms, subset of command driven systems and other applications where results can be easily examined without real-time interaction. Example

Like any other projects, web based projects also have users, requirements, schedules to be met and quality goals. Aspects of project management, requirements management, change control and quality management are applicable to web projects also.

The difficulty in fully specifying the requirements at the beginning of the project makes the conventional waterfall model unsuitable for web application development. In the traditional prototyping model, the prototype is essentially for capturing the requirements and is thrown away as soon as purpose is served.

(2) Projects that need better human – computer interfaces will benefit a lot from the prototyping

(3) Many engineering & scientific projects that involves large investments follow partying.

Problem:
What process adaptations are required if the prototype will evolve into a delivery system or product?

Answer:
Prototyping

Prototyping paradigm assists the developers and the stakeholders to better understand what to build when the requirements are fuzzy. Refer to figure 4.4 from the text book. Given below are the process adaptations techniques that are required when the prototype will evolve into a deliverable system or a product:

• The process must apply rigorous design rules from the start.

• Software Quality Assessments procedures must be applied from the beginning.

• The prototype must be designed with extensibility in mind and then it becomes the framework for extensions that will cause it to evolve into a production system.

Problem:
Provide three examples of software projects that would be amenable to the incremental model. Be specific.

Answer:
Software projects for incremental model

Incremental model:

The incremental model delivers a series of releases, called increments that provide progressively more functionality fc the customer as each increment is delivered.

The incremental model combines elements of waterfall model applied in an iterative fashion. Incremental model applies linear sequences in a staggered fashion as time progresses.

Incremental model is widely used to solve the problem of better integrating human factors into the system engineering and acquisition process. Examples:

1. Word processing software developed using incremental paradigm might deliver basic file management, editing an document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and grammar checking in third increment; and advance page layout capability in fourth increment.

2. Projects of type "Explore concept" are the province of the future studies group in marketing and strategic planning. Ex: Operating systems.

3. Projects of type "Proof of concept" are done by the engineering directorate.

Problem:
As you move outward along the spiral process flow, what can you say about the software that is being developed or maintained?

Answer:
Spiral model

A spiral model is combines the iterative nature of prototyping with the systematic control.

It uses the linear sequential model. It is defined by the software engineering team. The spiral model can be adapted t apply throughout the entire lifecycle of an application, from concept of development to the maintenance.

When an engineering team moves around the spiral, then the first circuit around the spiral results in development of product specification. The subsequent passes around the spiral might be used to develop prototype in more subsequent manner.

Problem:
Is it possible to combine process models? If so, provide an example.

Answer:
Combine Process Models

Yes, it is possible to combine the software process models.

Some possibilities to combine of software process models are given below,

1) Evolutionary process model.

2) Incremental process model.

3) The spiral model.In the evolutionary process models create gradually more complete version of software and a complete cycle of activities is repeated for each version.In the Incremental process model produces a series of release that provide more functionality for customer needs and increments are individually designed tested and delivered at successive points of time.In the spiral model produces the potential for fast development of progressively more complete versions of the software. It combines the features of the prototyping model and the waterfall model.Example projects:

Let us consider the case study of department of defense.

It follows waterfall model or an incremental model. But if we need to develop software based on some other software features, go with component based model. If time is the major factor apply rapid application model.

Hence we can combine evolutionary, incremental and spiral process models to develop single software. In medical and research projects also we can combine these process models.

Problem:
The concurrent process model defines a set of "states." Describe what these states represent in your own words, and then indicate how they come into play within the concurrent process model.

Answer:
Concurrent process model

It can be defined as a series of major technical activities, tasks, and their associated states. It is applicable to all types of software development process, and provides an accurate picture of the current stat of a project.

In this model, different parts of a project will be different stages of completeness those activities are all being performed concurrently. The activity for which initial communication was completed is said to be in none state. The challenge is to handle the concurrency about the project and be able to assess the status of it.

The stages included are as follows:

1) Under development

2) Awaiting changes

3) Under revision

4) Under review

5) Base lined

6) None

7) Done

These states represent the different stages of completion of the project, and are performed concurrently. It provides an accurate situation of the present state.

Problem:
What are the advantages and disadvantages of developing software in which quality is "good enough"? That is, what happens when we emphasize development speed over product quality?

Answer:
The advantages of developing software in which quality is "good enough" are

• Completeness – All the requirements are reflected in the software

• Conciseness – Compactness

• Reliability – No faulty outputs

• Efficiency – Amount of computing resources and cost required by a program to perform a function.

• Consistency.

We have to compromise with the advantages of quality software, where speed is major constraint but the speed has its own pros and cons.We have to compromise with the advantages of quality software, where speed is major constraint but the speed has its own pros and cons.

• Developing in short intervals resulting in miniature software projects and releasing the product in mini – increments but short iterations may not add enough functionally leading to significant delays in final iteration.

• Speedy software promotes strong collaborative atmosphere and dynamic gathering of requirements but dependency on strong cohesive teams and individual commitment to the project

Problem:
Provide three examples of software projects that would be amenable to the component-based model. Be specific.

Answer:
Commercial off – the shelf software components, developed by vendors who offer them as products , can be used when software is to be built. These components provide targeted functionality with well defined interfaces that enabl the component to be integrated into the software.

Regardless of the technology that is used to create the components, the component – based development model incorporates the following steps.

• Available component – based products are researched

• Component integration issuers are considered

• Comprehensive testing is conducted to ensure proper functionality.Example projects:

(1) All business application projects

Ex: Banking – operations like withdraw, deposit are same for any bank. So these can be developed into a component and can be reused.

(2) Airport traffic controllers

This project deals with traffic controlling in the airports.

Checking the flight arrival & departures.

(3) Many projects that use object – orientation paradigm are generally component

based.

Problem:
It is possible to prove that a software component and even an entire program is correct. So why doesn’ t everyone do this?

Answer:
Yes, it is possible to prove that a software component and even an entire program is correct, but it takes move time. Though the entire program is tested as we say that there is no 100% successful test, the program may go out of reacl some times. In order to make the entire program correct, we need to test the each & every possible test case but generally all the test cases may not be tested because some of the test cases may be unknown and the lock of time factor and the lack of coordination between the developers and the customers.Generally, instead of testing all the tes cases, only the frequent test cases that are being used by the customer are tested. This may give better results and even may gain customer satisfaction but the program may not be 100% correct. We cannot ensure that the software component and even the entire program is correct.

Problem:
Are the Unified Process and UML the same thing? Explain your answer.

Answer:
Unified process and UML

No, Unified process and the UML is not the same thing.

Unified process:

The Unified process is a type of framework, which is used for UML in software engineering. It is a popular iterative an incremental software development process which should be customized for specific organization or project.

Unified Modeling Language (UML):

engineering practice.

And also, UML is a graphical language for visualizing, specifying, and constructing the artifacts of software – intensive systemExplanation:

Both UML and unified process appears to be same thing on projects of all kinds. But there is a slight difference.

1. UML is applied in the unified process.

2. UML does not provide the process framework to guide project teams in their application of the technology.

3. Unified Process is a process framework in which UML may be applied as part of software engineering activities.

A model is a complete description of a system from a particular perspective. But the UML alone is not enough. A process defines Who is doing? What, When to do it and How to reach a certain goal

Example:

Agile Development is the process, it should use UML notation for produced diagrams as well."

Solutions: Chapter 4: Process Models.

4.1)

The waterfall model is amenable to the projects that focus on the attributes such as the data structures, software architecture, and procedural detail and interface characterization of objects.

4.2)

Software applications that are relatively easy to prototype almost always involve human-machine interaction and/or heavy computer graphics. Other applications that are sometimes amenable to prototyping are certain classes of mathematical algorithms, subset of command driven systems and other applications where results can be easily examined without real-time interaction. Applications that are difficult to prototype include control and process control functions, many classes of real-time applications and embedded software.

4.3)

If a prototype is evolved into a delivery system or product, it begins with communication. The software engineer and customer meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory. The prototype serves as a mechanism for identifying software requirements. If a working prototype is built, the developer attempts to make use of existing program fragments or applies tools (e.g., report generators, window managers, etc.) that enable working programs to be generated quickly.

4.4)

Each linear sequence produces deliverable "increments" of the software for example, word-processing software developed using the incremental paradigm might deliver basic file management, editing and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and grammar checking in the third increment, and advanced page layout capability in the fourth increment. The process flow for any increment may incorporate the prototyping paradigm. Incremental development is particularly useful when staffing is unavailable for a complete implementation by the business deadline that has been established for the project.

4.5)

As work moves outward on the spiral, the product moves toward a more complete state and the level of abstraction at which work is performed is reduced (i.e., implementation specific work accelerates as we move further from the origin).

4.6)

The process models can be combined, each model suggests a somewhat different process flow, but all perform the same set of generic framework activities: communication, planning, modeling, construction, and delivery/feedback.

of the efficiency of an algorithm, the adaptability of an operating system, or the form that human-machine interaction should take. In these, and many other situations, a prototyping model may offer the best approach. In other cases, an incremental approach may make sense and the flow of Spiral model may be efficient. Special process models take on many of the characteristics of one or more of the tradition.

4.7)

Stated simply, the concurrent process model assumes that different parts of a project will be different stages of completeness, and therefore, different software engineering activities are all being performed concurrently The challenge is to manage the concurrency and be able to assess the status of the project.

4.8)

The advantages of developing software in which quality is " good enough" is that the product or software will meet the deadline, it may however lead to the delivery of software that is low in quality and requires time to improper the quality. When speed is emphasized over the product quality it may lead to many flaws, the software may require mor testing, design and implementation work then done. Requirements may be poorly defined and may need to continuously change. Half hearted and speed may cause the risk management to fail to detect major project risks . Too little quality may result in quality problems and later rework.

4.9)

I would suggest postponing this problem until Chapter 30 is assigned, unless you do not intend to assign it. In that case, it can serve to introduce the importance of reuse.

4.10)

It is possible to use mathematical techniques to prove the correctness of software components and even entire programs (see Chapter 28). However, for complex programs this is a very time consuming process. It is not possible to prove the correctness of any non-trivial program using exhaustive testing.

4.11)

UML provides the necessary technology to support object-oriented software engineering practice, but it does not provide the process framework to guide project teams in their application of the technology. Over the next few years, Jacobson, Rumbaugh and Booch developed the Unified Process, a framework for object-oriented software engineering using UML. Today, the Unified Process and UML are widely used on OO projects of all kinds.

mike6606
followers - 0 followers - 5

0 0

« Previous: Software Engineering Research Methods for Practitioners Chapter 3 Answers
» Next: Software Engineering Research Methods for Practitioners Chapter 5 Answers

posted @ 2021-01-20 14:30 mike6606 read (971) Comments (0) Edit Favorite repor

Editor's Choice:
· Modern Image Performance Optimization - Fault Tolerance and Accessibility of Image Resources
· Talk: The Evolution of Service Architecture
· I'm on timeout with Redis again
· Clever use of CSS variables to create high-end grid animations
· Record the optimization of a lock

Reading Ranking:
· 3 mistakes to avoid when using Vue 10
· ASP.NET Core Web API interface throttling
· 【Fault Bulletin】CC attack is coming again, making March worse ·
Modern Image Performance Optimization and Experience Optimization Guide - Fault tolerance and accessibility of image resources
· How to JWT authenticate and authorize WebAPI in Net6.0