mike6606

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 Practitioner Research Methods Chapter 3 Answers

Problem:
In the introduction to this chapter Baetjer notes: “The process provides interaction between users and designers, between users and evolving tools, and between designers and evolving tools [technology].” List five questions that (1) designers should ask users, (2) users should ask designers, (3) users should ask themselves about the software product that is to be built, (4) designers should ask themselves about the software product that is to be built and the process that will be used to build it.

Answer:
(a) Users should ask designers

1. Will you tell me a little bit about yourself ?

2. How long have you been in business?

3. What your specialty ? How much does it cost?

4. What if I’ m not happy with the design?

5. Will you show me a few samples of your ideas for my project so I can get a feel for your work?

(b) Designers should ask users.

1. What are all of your needs for this project?

2. What is the timeline, deadline for this project?

3. Who will use this project?

4. What sites do you find similar?

5. Who will be your competitors?

(c) Designers should ask themselves.

1. What kind of development tools he / she uses?

2. How many applications he / she previously designed?

3. Does your web designer publish a blog?

4. Does all requirements reflect in the software?

5. What’ s the maintenance record?

(d) Users ask themselves.

1. How quickly developer understand my requirements?

2. Do I have budget to build it correctly?

3. Do I have the human resources to effectively maintain this functionality?

4. Does adding this functionality help me achieve my goals?

Problem:
Discuss the differences among the various process flows described in Section 3.1. Can you identify types of problems that might be applicable to each of the generic flows described?

Answer:
The software engineering process basically defines 5 framework activities. They are communication, planning, modeling, construction and deployment. They four types of process flow begin with communication activity.

Difference between the various process flows are described in section 3.1 are given in table below:-

	flow	flow	flow	flow
1	It executes the 5 framework activities in sequence.	It repeats a set of activities before proceeding to the next one.	It executes the activities in a circular routine.	It executes a set of activities in parallel with another activity.
2	It ends with deployment activity.	It ends with deployment activity.	After deployment activity it again starts with next circle for another version.	It ends with deployment activity.
3	There is a single version release.	There is a single version release.	Each circuit through the five activities leads to a more complete version of the software.	There is a single version release.
4	All 5 activities are gone through just once.	The first 4 activities can repeated.	The 5 activities can repeated a number of times	All 5 activities are gone through just once.
5	At a time only one activity is performed.	At a time only one activity is performed.	At a time only one activity is performed.	At a time one or more activity can be performed in parallel.

The problems that are faced by the generic flows described in section 3.1 are:-

- Linear process flow:

o Unpredicted integration issues can crop up.

o Defects can be traced to previous activities that is they are identified later.

o It is difficult for clients to state all requirements just initially.

- Iterative process flow:

o It can lead to a blocking state.

o More time is spent waiting than for production

- Evolutionary process flow:

o Project team waits for the earlier activity to finish.

o Developers can end up making compromises in implementation.

- Parallel process flow:

o It can cause confusion in the project team as it proceeds.

o It demands a good experts in risk assessment.

Problem:
Try to develop a set of actions for the communication activity. Select one action anc define a task set for it.

Answer:
A requirements gathering is an essential software engineering action for the communication activity.

A task set defines the actual work to be done to accomplish the objectives of a software engineering action.

The requirement gathering task set for communication activity follows:

- Create a list of stakeholders for the project.

- Interview each stakeholder individually to verify overall needs and requirements.

- Construct a preface list of features and functions based on stakeholder input.

• Generate informal user scenarios as part of each meeting.

• Improve user scenarios as a part of each meeting and the stake holder’ s feedback.

• Construct adjust list of stakeholder requirements.

• Apply quality function deployment techniques to prioritize requirements.

• Package requirements so that they can be delivered incrementally

• Note constraints and restrictions that will be placed on the system.

• Discuss method for validating the system.

Problem:

A common problem during communication occurs when you encounter two stakeholders who have conflicting ideas about what the software should be. That is you have mutually conflicting requirements. Develop a process pattern (this would be a stage pattern) using the template presented in Section 3.4 that addresses this problem and suggest an effective approach to it.

Answer:
When stakeholders have conflicting, the following process pattern describes an approach

about what the software should be

Pattern name: Conflicting Requirements

Intent: This pattern describes an approach for identifying and making a list of the requirements specified by the stakeholders.

Type: Stage pattern.

Initial context:

The following conditions must be met prior to the initiation of this pattern.

1. Stake holders have been identified;

2. A mode of communication between stakeholders and the software team has been established;

3. The overriding software problem to be solved has been identified by stakeholders;Problem: Requirements of the two stakeholders are conflicting. There is a clear recognition of the requirements but, the ideas of the stakeholders ar not the same. No clear idea of what the software should be.

Solution: Evolutionary process models are used to solve this problem. The evolutionary process models recognize the iterative incremental nature of the project and are designed to accommodate the changes. As the requirements are conflicting, a preliminary list of functions and features based on stakeholder input is built. The pattern is built according to the basic requirements. And then the changes are accommodated using the evolutionary models.Resulting context: A software prototype that identifies basic requirements is approved by the stakeholders. The prototype is built and changes are made, if needed. Or the pattern will be rejected and a new pattern with new features is developed.

Related patterns:

Customer Communication,

Customer Assessment,

Requirements Gathering,

Constraint Description.

Known uses and examples. This pattern is needed when the requirements are uncertain and conflicting."

Solutions: Chapter 3: Software Process Structure.

3.1)

a) Designers should ask users:

Is the product satisfactory, or does it require redesign or rework?

Was user input solicited, to avoid the product being unsatisfactory and requiring

Is the product larger than estimated?

Do the modules require more testing, design and implementation work to correct

than expected?

b) Users should ask as designers:

Is the scope clear?

Do we have the tools and people with skills required for the development?

Are the requirements properly defined, are additional requirements needed.

Are the specified areas of the product more time consuming than usual?

Does the module require more testing, design?

c) Users should ask themselves about the software product that is to be built:

What is the scope and purpose of the software product?

Is the product larger than estimated?

Are the best people available?

Is the staff committed and possess skills required?

Will the turnover among staff members be low enough to allow continuity?

d) Designers should ask themselves about software product that is to be built and the process that will used to build it:

Scope and purpose of the document?

What tools are to be used?

What are the objectives and risk aversion priorities?

What will be the steps for risk analysis, identification, estimation, and evaluation and management?

3.2)

1. Linear process flow does not accommodate change well, but can be good if a team is building a routine product similar to something they have done before
2. Iterative process flow handles change better by building in opportunities to reviews the intermediate work products as they are developed. Often used when building systems involving technologies that are new to the development team.
3. Evolutionary process models are often adopted for projects (e.g. WebApps) that need to be developed in a rapid, but controlled manner that avoids unnecessary rework.
4. Parallel process flow has the potential to allow self-contained work products to be developed simultaneously for systems that are composed of subsystems.

3.3) Task Set for Communication Activity: A task set would define the actual work to be done to accomplish the objectives of a software engineering action. For the communication activity these are:

Make a list of stakeholders for the project

Invite all the stakeholders to an informal meeting

Ask them to make a list of features and functions

Discuss requirements and build a final list

Prioritize requirements and note the areas that he is uncertain of

These tasks may be larger for a complex software project, they may then include

To conduct a series of specification meetings, build a preliminary list of functions and features based on stakeholder input.

Note constraints and restrictions on the system.

Discuss methods for validating system.

3.4)

> Pattern Name. *Conflicting Stakeholder Requirements*
>
> Intent. This pattern describes an approach for resolving conflicts between stakeholders during the communication framework activity.
>
> Type. Stage pattern
>
> Initial context. (1) Stakeholders have been identified; (2) Stakeholders and software engineers have established a collaborative communication; (3) overriding software problem to be solved by the software teams has been established; (4) initial understanding of project scope, basic business requirements and project constraints has been developed.
>
> Problem. Stakeholders request mutually conflicting features for the software product under development.
>
> Solution. All stakeholders asked to prioritize all known system requirements, with resolution being to keep the stakeholder requirements with highest priorities and/or the most votes.
>
> Resulting Context. A prioritized list of requirements approved by the stakeholders is established to guide the software team in the creation of an initial product prototype.
>
> Related Patterns. Collaborative-guideline definition, Scope-isolation, Requirements gathering, Constraint Description, Requirements unclear
>
> Known Uses/Examples. Communication is mandatory throughout the software project.

Good text to top Follow me
Bookmark this article

mike6606
followers - 0 followers - 5 0 0

+ Plus follow

« Previous: Software Engineering Research Methods for Practitioners Chapter 2 Answers
» Next: Software Engineering Research Methods for Practitioners Chapter 4 Answers

posted @ 2021-01-20 14:14 mike6606 read (951) Comments (0) Edit Favorite repor

Refresh Comment Refresh page returns to the tc

Log in to view or post comments, log in now or browse the blog home page

Editor's Choice:
· Modern Image Performance Optimization - Fault Tolerance and Accessibility of Image Resources
· Talk: The Evolution of Service Architecture

· 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