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 5 Answers

Problem:
Reread the "Manifesto for Agile Software Development" at the beginning of this chapter. Can you think of a situation in which one or more of the four "values" could get a software team into trouble?

Answer:
Manifesto for agile software development:

Better ways of developing software are as follows:

1. Individuals and Interactions over Processes and Tools:

It is the first value proposition of the manifesto. In this tool should be used for capturing requirements, version control, project planning, editing, file transfer, etc. Here the process can get in the way, and it is more important to ge to the point which is using the interaction over process and will get you much better results.

2. Working Software over Comprehensive Documentation:

This tells about the importance of documentation, which is often lacking in projects and can cause problems.

It is not important, but, when it comes down to a choice between finishing a project with a working outcome. It is really important to the client.

3. Customer Collaboration over Contract Negotiation:

The purpose of contract negotiation is to ensure that responsibilities and deliverables are clear, as well as providing a framework for resolving issues should they arise. Relationship with the client are able to resolve things amicably, a contract is more like an insurance policy should things go wrong.

4. Responding to Change over Following a Plan:

The purpose of the plan is to know what you are deviating from when you have to make a change. Change is inevitable, there's no point in fighting it.

Creating the initial project plan is an important process to capture all the tasks required, the ideal order in which they should be done, and most importantly, to identify dependencies. However, the initial project plan and how things happen in reality are often quite differentDefinitely one or more of the four "Values" could get a software team into trouble.

Example:

The customer, programmer interaction may not go well, the user may not specify all the requirements that the software must under go. There may be a misunderstanding in the interaction between customer and the developer. In such situations the software teams automatically creeps into troubles.

So, from the fourth value, the software team will get into trouble.

Agile process models have been designed to address the following issues:

1. Importance of self-organizing team that has control over work they perform.

2. Communication and collaboration between team members, practitioners and their customers.

3. An emphasis on rapid delivery of software that satisfies the customer.

Problem:
Describe agility (for software projects) in your own words.

Answer:
The word agility means "To do something very quickly". Thus, agility in terms of software development is to develop a software which meets all the requirement specified by the customer, and always ready to accept the changes required by the customer even in later phases. The goals of agility are explained below:

• The development process should always be ready to accept new changes needed in the development.

• Agility means to deliver the working software as soon as possible.

• Agile means to keep the software simple, the part only that is required.

• Agility in the software development keeps the change cost low as it is an incremental process. thus, changes can be easily made in controlled environment.

Problem:

Why does an iterative process make it easier to manage change? Is every agile process discussed in this chapter iterative? Is it possible to complete a project in just one iteration and still be agile? Explain your answers.

Answer:
The software team manages change by focusing on a defined increment and postponing any changes until the next increment. All agile process models are iterative/incremental.

An iterative process make it easier to manage changes. Since each iteration is a mini – project, the project team addresses, to some extent, all the risks associated with the project as a whole each time it builds an increment of the system. As risks become greater, as delays occur, and as the environment become more unstable. The team is able to make necessary adjustments on a relatively small scale and propagate those adjustments across the entire project. Ar iterative process has greater flexibility to change the plan. Hence it is easier to manage the changes. Almost all the agile process models are iterative.

Yes, it is possible to complete a project in just one iterative and still be agile. Because if the time is the major constraints and all the requirements are well known at the beginning of the project. If there are well suited design patterns are available and the principles of agile development are satisfied then it is possible to complete a project in one iteration that too with agility.

Problem:

Could each of the agile processes be described using the generic framework activities noted in Chapter 3? Build a table that maps the generic activities into the activities defined for each agile process.

Answer:
The generic process framework can be applied to each agile process described in this chapter. The table should list al agile process models across the first row and all generic framework activities

• Communication

• Planning

• Modeling

• Construction

• Deployment.Mapping from activates in agile process to generic activities.

• Extreme programming (XP)

o Planning – communication, planning

o Design – modeling

o Coding – modeling, construction

o Test – deployment.

• Adaptive software development (ASD)

o Speculation – communication, planning

o Collaboration – modeling, construction

o Leaning – Deployment

• Dynamic systems development method (DSDM)

o Feasibility, business study – communication, planning

o Functional model, design – Modeling, constructs

o Implementation – construction, deployment.

Problem:

Try to come up with one more "agility principle" that would help a software engineering team become even more maneuverable.

Answer:
One more additional agility principle –

you know how and by whom they will be used.Travel light and update only when it hurts. Everything that is created, model, document, code, test etc has a cost. If you decide to keep it, each will need to be maintained over time increasing cost of change and what people must know and understand. The less you keep and maintain, the more agile you are.

Minimize the complexity. Simplicity reduces amount of work to do. The simplest solution is usually the best solution. More complex and detailed artifacts will be more difficult and expensive to change. Plan for change realistically, but don’ t overload your models or system with features that your don’ t need.

Problem:

Select one agility principle noted in Section 5.3.1 and try to determine whether eacl of the process models presented in this chapter exhibits the principle. [Note: We have presented an overview of these process models only, so it may not be possible to determine whether a principle has been addressed by one or more of the models unless you do additional research (which is not required for this problem).]

Answer:
Agile processes embrace change as essential to the customer's competitive advantage. Each of the process models presented in this chapter exhibits this principle.

Selected one agility principle is: “Working software is the primary measure of progress”.

1. Extreme programming (XP):

Planning activity begins with the creation of a set of stories. A value is assigned to each story and XP team will assess each story & assign a cost. Finally the XP story will evolve as working software.2. Adaptive software development (ASD):

Effective collaboration with the customer will only occur if we throw out any “us and them” attributes. ASD teams with intent of learning & then improving its approach.3. Dynamic system development method (DSDM):

It is an incremental method. The increment may not be 100% complete but there should be some progress comparec to last increments4. Scrum:

Scrums allow us to build softer software. Scrum patterns enable a software development team to work successfully in a world where elimination of uncertainty is impossible.5. Agile modeling (AM):

The problem can be partitioned effectively among the people who must solve it and quality can be assessed at every step as system is being progressed, engineered and built.

Problem:

Why do requirements change so much? After all, don’ t people know what they want?

Answer:
Requirements may change for various reasons even if people know their wants. The reasons for the same can be:-

• Change in environment: Changes in legislation, organizational dynamics or competition strategy may also lead to change in requirements.

• Missed requirement: A stakeholder working with a system might realize that it's missing a feature.

• Identify a defect: A bug or defect in the system, creates new requirement.

• Actual requirement was not clearly understood: It is common for user to realize that what they asked for is different from what is required.

It is a fact that requirements change a lot in software development life cycle. Changes in requirement are hard to predict and so are the change in customer priorities.

It is often difficult for stakeholders to verbalize their software needs till they see a working prototype, and it is then only that they might realize that they have forgotten to consider some important points. This might happen because people may not know what they actually want until they see something running. This explains importance of active stakeholder participation and short iterations to a system’ s success.

Very rarely, does it happen that stakeholders do not know what they want from the system. A decent requirement specification can be designed by effective collaboration between all the stakeholders, also their experience in the respective fields does make the task a little easier.

Problem:

Most agile process models recommend face-to-face communication. Yet today, members of a software team and their customers may be geographically separated

Answer:
Face – to – force communication is a better way to collect the requirements from a customer, but face – to – face communication leading to many interpersonal difficulties and miscommunications in today’ s workplace. Major problem in today‘ s software development is that the members of software team and their customers may be geographically separated from one other. The ways to overcome this problem is to use

E – mails, text messaging, PDA’ s, cell phones, video conferencing, black berries, blueberries, raspberries, and more.

It’ s unbelievable how dependent we‘ ve become as a society on electronic communication devices!

Problem:
Write an XP user story that describes the “favorite places” or “favorites” feature available on most Web browsers.

Answer:
Internet bookmarks are stored webpage locations that can be retrieved. The main purpose is to easily catalog and access web pages that a user has visited and choose to save. Saved links are called “favorites”, and by virtue of the browser‘ s large market share, the term favorite has been synonymous with bookmark since the early days of widely – distributed browsers. Bookmark are normally visible in a browser menu and stored on the user‘ s computer and commonly a metaphor is be used for organization.Book marks are a fundamental feature of web browsers, but some users have expressed frustration with bookmark collections that become disorganized and have looked for other tool to help manage their links. There tools include browser synchronizers and desktop applications.

Problem:
What is a spike solution in XP?

Answer:
Extreme Programming (XP):

• Extreme programming (XP) is a popular agile software development methodology used to implement the software project.

• XP uses an object-oriented approach as its preferred development paradigm.

• XP encompasses a set of rules and practices that occur within the context of framework activities.Spike solution:

• It is a small technique to investigate the solution to a problem.

• To figure out tough problems or any other design problems, create a spike solution to overcome it.

• When we encounter a tough problem that lack’ s an immediately apparent solution, create a spike solution.

• The goal of a spike solution is to keep the solution simple and away from the risk factors.

• That is where you try out different approaches in the code to make the correct solution more apparent.

• This happens all the time in independent development, and it may not require the level of formally those agile programming calls for.Spike solution in XP:

A spike solution in XP is a prototype that is intended to throw away.

• The prototype is implemented and evaluated. The intent is to lower risk when true implementation.

• Starts and to validate the original estimates for the story containing the design pattern.

• To figure out the tough problems or any other design problems, create a spike solution to overcome it.

• Also, the spike solution reduces the potential risk of the problem.

• The spike approach deals to an “end to end” approach to the problem.

• In XP, the spike solution implemented to keep away the risk and to prepare the quality solutions to the tough problems when those agile programming calls for.

Problem:
Describe the XP concepts of refactoring and pair programming in your own words.

Answer:
Refactoring:

It is a construction technique. Extreme programming (XP) encourages refactoring. Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves the internal structure.. it is a disciplined way to clean up the code that minimizes the chance of introducing bugs. The intent of refactoring is to control the modifications by suggesting small design changes that can radically improve the

Pair programming recommends that two people or teams work together at one computer work station to create cod for a story. This provides a mechanism for real – time problem solving and real – time quality assurance. In practice, each person takes on slightly different role. For example, one might think about coding details of particular portion o design while the other ensures that coding standards are being followed and code that is generated will fit to requirement.

Problem:
Using the process pattern template presented in Chapter 3, develop a process pattern for any one of the Scrum patterns presented in Section 5.5.1.

Answer:
Process pattern template for scrum pattern "communicate Early"

Problem: What is the goal of a project and who are members of a team?

Context: Use of scrum in a distributed project (faster, cheaper and quality projects)Solution: Arrange kick – off meetin for all relevant members present briefly scrum methodology.

Describe the goal and contents of the project.

Describe a release plan with some sprints

Introduce briefly use of an ALM tool everybody presents his / her responsibilities & hobbies.Consequences: common goal is known by every relevant member a common process and

tools for every site can be established by scrum and ALM tools efficient

communication channels can be created between team members when

they learn to know each other better.

Problem:
Visit the Official Agile Modeling site and make a complete list of all core and supplementary AM principles.

Answer:
Core principles of agile modeling:

• Assume simplicity

• Embrace change

• Enabling the next effort is your secondary goal

• Incremental change

• Maximize stakeholder ROI

• Model with a purpose

• Multiple models

• Quality work

• Rapid feedback

• Working software is your primary goal

• Travel lightSupplementary principles of agile modeling:

• Content is more important than representation

• Open and honest communicationAssume simplicity:

• Develop the model as simple as possible. Avoid including additional features that are not necessary.

Embrace change:

• Requirements of the user changes from time to time. Therefore, the agile model should be willing to accept the change.

Enabling the next effort is your secondary goal:

• The primary goal of software development is develop software that is working efficiently and meets all the requirements of the customer.

• Sometimes even though working software is delivered, it is still considered as a failure.

• The secondary goal is the next effort to develop a system with next major release.

• It is not mandatory that the model must be perfect the first time it is built.Maximize stakeholder ROI:

• Consider the resources that are provided by the stakeholders while designing a model so that stakeholder ROI is maximized.

Model with a purpose:

• A model should be developed so that it will give a better understanding of the product to be designed and help in reducing the complexity of the product to be designed.

Multiple models:

• If the software requires multiple models, then develop models that are necessary for the software.

• Not all products require multiple models.

Quality work:

• Develop the model that is of good quality.Rapid feedback:

• Rapid feedback can be obtained when working closely with customers.

• It helps in understanding the requirements of the customer.

• The customer can give a feedback after the user interface has been developed.

Working software is your primary goal:

• The primary goal of software development is develop software that is working efficiently and meets all the requirements of the customer.

Travel light:

• Develop the model as simple as possible so that when a change has to be made, it will be done easily.

• If multiple models are maintained, then a change has to be reflected in all the models. So maintain simple models that are necessary.Content is more important than representation:

• The type of model or the representation of the model constructed is not important.

• The model should be developed so that it serves its purpose.

Open and honest communication:

• People should be open and honest to voice up their ideas or opinions or suggestions.

Problem:

The tool set proposed in Section 5.6 supports many of the "soft" aspects of agile methods. Since communication is so important, recommend an actual tool set that might be used to enhance communication among stakeholders on an agile team.

Answer:

Solutions: CHAPTER 5: AGILE DEVELOPMENT

5.1) One situation in which one or more of the four "values" could get a software team into trouble would be Responding to change over following a plan, In many situations, we no longer are able to define requirements fully before the project begins. Software engineers must be agile enough to respond to a fluid business environment or else they might land up in trouble.

5.2) Agility can be applied to any software process. However, to accomplish this, it is essential that the process be designed in a way that allows the project team to adapt tasks and to streamline them, conduct planning in a way that understands the fluidity of an agile development approach, eliminate all but the most essential work products and keep them lean, and emphasize an incremental delivery strategy that gets working software to the customer as rapidly as feasible for the product type and operational environment.

software increment. For example, ASD uses an iterative process that incorporate adaptive cycle planning, relatively rigorous requirement gathering methods, and an iterative development cycle that incorporates customer focus groups and formal technical reviews as real-time feedback mechanisms. The Dynamic Systems Development Method (DSDM) defines three different iterative cycles—functional model iteration, design and build iteration, and implementation— preceded by two additional life cycle activities—feasibility study and business study.

5.4) Answers will vary

5.5) One more "agility principle" that would help a software engineering team become even more maneuverable would be, "A team should know whose skills suit a particular project, and get these people on their project, for software development to become more effective", and " Communication is the key, the consumer and developer should constantly be communicating even if they are geographically separated, they can web talk".

5.6) An agility principle noted in Section 5.3.1 is "Welcome changing requirements, even late in development". Agil processes harness change to the customer's competitive advantage, each of the process models presented in this chapter exhibits this principle.

5.7) Requirements change so much, that it is difficult to predict in advance which software requirements will persist and which will change. It is equally difficult to predict how customer priorities will change as the project proceeds. It i often difficult for people to verbalize their software needs until they see a working prototype and realize that they ha forgotten to consider something important.

5.8) Most agile process models recommend face-to-face communication. Yet today, members of a software team and their customers may be geographically separated from one another in that case Communication is the key, the consumer and developer should constantly be communicating even if they are geographically separated, they can webtalk or talk over the phone every now and then or write emails, use chatting as the, means or a medium of conference call communication where 2 and more people can talk to each other at the same time.

5.9) Answers will vary

5.10) If a difficult design problem is encountered as part of the design of a story, XP recommends the immediate creation of a operational prototype of that portion of the design, called a spike solution, the design prototype is implemented and evaluated. The intent is to lower risk when true implementation starts and to validate the original estimates for the story containing the design problem.

5.11) XP encourages refactoring—a construction technique that is also a design technique. Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves the internal structure. A central notion in XP is that design occurs both before and after coding commences. Refactoring means that design occurs continuously as the system is constructed a key concept during the coding activity is pair programming. XP recommends that two people work together at one computer workstation to create code for a story. This provides a mechanism for real-time problem solving (two heads are often better than one) and real-time quality assurance.

5.12) The DSDM approach for each iteration follows the 80 percent rule. That is, only enough work is required for eac increment to facilitate movement to the next increment. The ASD time-boxing approach establishes distinct schedule boundaries for the delivery of components of an increment and suggest that work continues only until the time-box boundary is reached. The remaining detail can be completed later when more business requirements are known or changes have been requested and accommodated.

5.13) Answers will vary

5.14) One web site to examine for AM principles might be http://www.agilealliance.org/home

5.15) Collaborative software engineering tools like Enterprise Architect paired with Microsoft Visual Studio might be examples of real tools that support (along with version control, bug tracking software, and instant messaging)

mike6606
followers - 0 followers - 5

0 0

« Previous: Software Engineering Research Methods for Practitioners Chapter 4 Answers
» Next: Software Engineering Research Methods for Practitioners Chapter 6 Answers

posted @ 2021-01-20 14:36 mike6606 read (1473) Comments (0) Edit Favorite repor