A’Kaia
Natasha
Hannah

CMSI 4071: Pilone & Miles Book Assignment

Assignment 2 Description

You should already have read at least the first three reading assignments so that you are familiar with the topics. Once you have read these reading assignments, answer the following questions. Remember, **all homework must be typed**.

Due date: Week 9, beginning of class

Problem 1

Write a short paragraph to answer these three questions:

- What are the two major concerns of any software project?
- Which of those two do you feel is more important?
- Where does the idea of complete functionality fit with these two concerns?

Two major concerns of any software project are delivering on time and staying on budget and delivering the customers needs with working software. Both are very important but delivering working software that meets the customers expectations is more important because if it doesn't work even if it's delivered on time, it's useless. Complete functionality is the balance staying on time and within the budget with the goal of delivering a functional project that satisfies the customer.

Problem 2

Write a short paragraph to answer these three questions and briefly explain your opinion:

A’Kaia
Natasha
Hannah

- In the Agile method for software development, what are the five main phases that occur in each and every iteration?
- Do you feel that any of them could be done at the start of the project and not be repeated in every iteration?
- Do you feel that would save time overall on the project?

Five main phases in each and every iteration are requirement gathering, design, implementation, testing, and review. No, I think these steps are repeated for a reason, things change during iterations. Not repeating wouldn’t save time because the end result wouldn't align with what the customer wanted due to miscommunication.

Problem 3

Write a short paragraph to answer these four questions and briefly explain your opinion:

- In the Waterfall method for software development, what are the main phases that occur?
- How are they different from the phases in the Agile method?
- What other phases are in Waterfall that are left out of Agile?
- Do you think these are needed in Waterfall?
- Describe a situation using Agile in which one of these extra Waterfall phases might be needed.

In Waterfall, the main phases are requirements, design, implementation, verification, maintenance. Waterfall completes each phase before moving onto the next and tends not to go back to completed phases unlike Agile. Waterfall phases that Agile doesn't have are documentation, approval gates, and maintenance phases. I do think they are needed in Waterfall. In Agile, it might need an approval gate if there's a legal requirement with the customer.

Problem 4

Write one-sentence answers to the following questions:

- What is a user story?

A’Kaia
Natasha
Hannah

A user story is a short description of a feature or functionality told from the perspective of the user to capture what they want and why.

- What is blueskying?

Blueskying is the process of brainstorming unique, unrestricted ideas without considering limiting factors like practicality in order to inspire innovation and new ideas for future use cases.

- What are four things that user stories SHOULD do?

User stories should be concise, clearly communicate the needs of the user and why, be a collaboration between stakeholders and developers, and focus on delivering tangible value to the end user through clear, actionable outcomes.

- What are three things that user stories SHOULD NOT do?

User stories should be concise, clearly state the user’s need and goal, be written collaboratively with the customer, and guide development toward user value.

- Does the Waterfall method have user stories?

No, the Waterfall method uses fixed, detailed requirements up front rather than flexible, evolving user stories.

Problem 5

What is your opinion on the following statements, and why do you feel that way:

All assumptions are bad, and no assumption is a good assumption.

I agree because assumptions often lead to misunderstandings and wasted work. The book says to put assumptions “on trial for their lives,” meaning we should question and confirm them with the customer instead of guessing.

A'Kaia
Natasha
Hannah

A big user story estimate is a bad user story estimate.

I agree because large stories are too vague to estimate or complete accurately. Breaking big stories into smaller ones makes planning clearer and development more manageable.

Problem 6

Fill in the blanks in the statements below, using the following things [you can use each thing for more than one statement]: Blueskying; Role playing; Observation; User story; Estimate; Planning poker.

- You can dress me up as a use case for a formal occasion:

  User story____________________________________

- The more of me there are, the clearer things become:

  User story____________________________________

- I help you capture EVERYTHING:

  Blueskying, Observation____________________________________

- I help you get more from the customer: role playing, observation____________________________________
- In court, I'd be admissible as firsthand evidence: observation____________________________________
- Some people say I'm arrogant, but really I'm just about confidence: estimite____________________________________
- Everyone's involved when it comes to me: Blueskying____________________________________

NOTE: when you have finished, check your answers with the result in your text on page 62. Do you agree with the book answers? If you disagree with any of them, justify your preferred answer.

A’Kaia
Natasha
Hannah

Problem 7

Explain what is meant by a better than best-case estimate.

- A “better than best-case” estimate ignores risks and uncertainty, leading to missed deadlines and disappointed customers.

Problem 8

In your opinion, when would be the best time to tell your customer that you will NOT be able to meet her delivery schedule? Why do you feel that is the best time? Do you think that would be a difficult conversation? If so, how could you make it less difficult?

- In my opinion, the best time to tell the customer you will not be able to meet the delivery schedule is as soon as you realize the deadline is at risk.

  Telling them early allows time to adjust priorities, drop lower value features, or extend the timeline before it becomes a crisis. Waiting until the deadline is near damages trust and limits options.It would definitely be a difficult conversation because no customer wants to hear about delays and it is a very confrontational conversation for both parties. To make it less difficult I would definitely turn it into more of a problem solving moment then a blame moment.

Problem 9

Write a short paragraph to discuss why you think branching in your software configuration is bad or good, then describe a scenario to support your opinion.

We think branching is good because it helps developers fix bugs without interfering with the main code base. If a team is building a mobile app, one developer can make a branch dedicated to creating dark mode and they can safely test their code.

A’Kaia
Natasha
Hannah

If theres too many branches, its more likely to have merge conflicts and if developers delay merging, the code will be outdated. If a team is rapidly changing their project and the developers have their own branches, there will likely be merge conflicts due to the rapid changes which leads to time wasted trying to solve errors

Problem 10

Have you used a build tool in your development? If you have, which tool have you used? What are its good points and bad points — in other words, what do you like about it and/or dislike about it?

Yes, we’ve used npm and Xcode build system and it's nice because it compiles and tests for you which makes collaborating easier since everyone uses the same commands. Downside is errors can be tricky to understand since there isn't always a lot of detail.