Laura Berrios, Bikram Gahley, Mohammad Hasan, AbdulBasit Jeewa, Philippe Labruyere, Weiyi Liu, Brian Salinas

Before doing this assignment please note this is not a copy and paste exercise. Doing so will result in a failing grade and you will be reported for plagiarism. This is an assignment that involves research and writing, in your own words. Spelling, grammar and punctuation mistakes will impact your grade.

Part A:

Watch the following You-Tube videos

1. Seven Reasons for Agile: Reason No: 4

Reviewing the working software is more important than reviewing a bunch of requirements specification documents.
When the requirements are not fully known in the beginning of the project, it is not practical to expect the customers to review and approve the specifications in the beginning.

https://www.youtube.com/watch?v=GAr-tvN7QBQ&list=PL9MIlke8ddNt_bE4phBFuncI5BibC9S0_&index=6

2. Why Agile? Seven Reasons for Agile Software Development: Reason No: 5

The best way to develop software is to build in small increments, based on known requirements.
You need to collect new requirements based on what is already known, progressively.
Agile does this type of iterative and incremental software development, which adds a lot of value to the customers.

https://www.youtube.com/watch?v=w48nZQdWEWI&t=59s

3. Why Agile? Seven Reasons for Agile Software Development: Reasons 6 & 7: and conclusion

Here are the last two reasons for Agile Software Development.

https://www.youtube.com/watch?v=njApjmV9DM0&list=PL9MIlke8ddNt_bE4phBFuncI5BibC9S0_&index=8

Part B:

Based on the You-Tube video, answer the following questions.

1. According to Raj, requirements changes are inevitable in Software Development. One of the major problems of traditional software development is change resistance mentality.
   Explain why customers may feel uncomfortable signing off on waterfall requirement specifications?
   Customers are not 100% certain that they know exactly what they want for

their project. At any point during the development phase, the customer could think of another requirement but since waterfall methods expect all requirements to be specified from the very beginning, they are unable to change the requirements. Ideas can pop up at any moment, which is why customers are uncomfortable signing off on waterfall requirement specifications.

2. How does agile methodology differ from waterfall in how requirements are gathered?
In Waterfall, all requirements are determined and gathered at the start of the project. In Agile, on the other hand, while requirements are still gathered at the start, they can be changed and added to later if the need arises.

3. Waterfall is sequential development with one big deliverable and agile is iterative and incremental software development with incremental with several minimum viable product (MVP) deliverables. Explain how producing incremental MVP deliverables help the client eventually get what they want in a finished product?
In Waterfall, the customer is only involved at the beginning of the project and is expected to give all of the requirements before the project begins and cannot change those requirements once the development has begun. In Agile, the customer is involved in the project every step of the way and is welcome to change the requirements as the development goes on. With just one final deliverable from a development process where the customer was not involved, the customer can be unhappy with the final product and will essentially be stuck with an unsatisfying product or lose money by hiring a new team and starting over from scratch. With minimum viable deliverables, the customer can test the product during each sprint and is more likely to be satisfied with the final product. Agile provides clients to make changes in between sprints and lead the product along a path that they will be more satisfied with.

4. Big, upfront planning is not practical in software development. When the requirements are not fully known in the beginning of the project, the effects of that will reflect on the planning. How does Just In Time (JIT) planning help agile project teams manage their projects?
Just-In-Time planning in agile is used in almost every sprint of Agile. JIT tells a project in agile to move forward. This is because a team doesn't plan 100% of the project in one sprint. They plan "just enough" work to progress the project forward, and no more.

5. In the discussion of why only 16% of all software projects are successful, Raj mentions the Standish report that says that user involvement contributed the highest to project success. How does Agile differ from waterfall in user (client) involvement?
In Agile, the client is part of the team and is involved throughout the process.

In Waterfall, on the other hand, the customer is distanced from the process.

6. Compare and contrast the difference on how each methodology, agile and waterfall use reflection to improve their processes.
   In Agile, a team can reflect at the end of each sprint and use the lessons learned in that same project. In contrast, Waterfall does not allow for a team to apply any lessons learned from reflecting on a project in that same project, as it will be too late to go back to an earlier phase of said project.

Part C:

Based on assigned readings, work as a group to answer the following question:

(1) Discuss as team and write down 5 traits of poisonous person as described by our textbook.

Ego, Lack of Respect for Other People's Time, Entitlement, Confusing communication, and Rudeness.

(2) Discuss as team and write best practices for identifying a threat from poisonous person as described by our textbook.

Look to see if a team member's behavior is disrupting the team's workflow. One way to do this is to listen for complaints about team members' behavior. A team should make sure they are communicating with each other properly and positively.

(3) Our textbook author suggests repelling trolls with niceness. Discuss this concept and come to a contentious about how your team feels about this construct. Do you agree or disagree and why?

Treating trolls with niceness should definitely be the first resort for dealing with them.

(4) Is there ever a time you should give up trying to be nice?

If being nice fails to adjust the toxic team member's problematic behavior, then the team should take a different approach to deal with the toxic team member.

Part D:

Have the following discussion about improving team communication:

Was everyone prepared for class?
Was everyone engaged in the discussion?
Did everyone contribute equally this class?
Was there one person who monopolized the discussion?

What will you do as individuals and as a team to address these issues and improve team communication?

Several team members had not read the correct chapter of the reading, but everyone caught up in time to do the work. The entire team participated in the discussion. The entire team participated approximately equally. The discussion was not monopolized by any team member.