Brandon Briggs

Brendan Noall

Alejandro Marcovich

Mark Powers

HW #4

Chapter 7

**1.) Why is it better to find defects earlier rather than later in the product development cycle?**

Implementing a continuous focus on defect detection earlier rather than later is better to avoid delay in production delivery. If a bug is found after delivery, there is risk of losing product reputation and the loss of valuable customers.

**3.) How does test-driven development improve code quality?**

Test-driven development (TDD) improves code quality because it gives the developers the mindset to write code essential to pass the tests. To do this, the developers first develop the test cases and then think through the requirements before writing the code.

**5.) What is the benefit of decoupling code?**

The benefit of decoupling code is to reduce the number of changes that are required if you want to use a different module. This would reduce rework required since each piece is dependent on each other.

**6.) What risk may you encounter when doing regression testing?**

Regression testing does not cover all test cases and cannot detect system-level problems.

**9.) What is the difference between acceptance and system testing?**

The main difference between acceptance and system testing is that acceptance testing focuses on performing tests on the product to ensure the product meets the agreed-upon requirements while system testing would test the system as a whole emulating the customer experience.

**11.) How is customer testing different in Agile development environments?**

Customer testing is different in Agile development environment because it gets the customer intimately involved with the product development team rather than focus on the terms and conditions of the project. The development team may initiate this level of involvement through the customer council.

**Chapter 8**

**1.) How is Kanban different from the other Agile methodologies, specifically Scrum?**

Kanban is different from other Agile methods because it doesn’t assign time-boxes for development. Kanban utilizes a ‘continuous flow’ model, meaning that tasks are continuously added to backlog and removed as they are completed. When compared to Scrum, Kanban is better for unpredictable workflow, unplanned work, and development tasks. Scrum is better for time-boxed workflow such as product development efforts, that can be broken into a series of sprints.

**2.) What do the columns on a typical Kanban board represent?**

Columns on a typical Kanban board represent the software development workflow, which is typically in the following order:

1) Backlog: All user stories listed in order of priority

2) Not Started: Stories are selected by developers (assigned), similar to ‘pending’ status

3) In Progress: Currently in work

4) Testing: To ensure code works as intended

5) Completed: When stories are considered ready to be delivered

**6.) What are information radiators, and how do they help Agile teams?**

Information radiators are anything posted in the team members' physical space that they will walk by or see on a regular basis. The concept is that as you walk by them and see them over-and-over again, they will become part of your subconscious like a billboard you drive by on the freeway.

**7.) What is the difference between a burn-up chart and a burn-down chart?**

Burn-up charts show progress toward a products release goal; whereas, burn-down charts provide daily status checks for the team relative to where they expected to be at a point in time, which helps in determining if course corrections are needed during a given sprint.

**13.) What does the daily stand-up meeting provide to the team beyond simply giving status?**

The daily standup asks 3 key questions outside of just ‘giving status’, which provides a medium for collaboration:

1) What did I do yesterday: provide info on completed tasks

2) What am I planning to do today: Provides insight into what the developer is working on today

3) Is there anything blocking my progress: Anything from a technical issue to a quick clarification question to the product owner.

**15.) Why is it important for a team to hold a retrospective session?**

Retrospectives are a critical component to the Agile development process. It is at the end of an iteration when the team looks back on the progress they made and discuss how they can be more effective. Open communication among members then results in fine tuning and adjustments to the team's behavior as needed. This willingness to adapt is important for improving workflow.

**16.) What is the best indicator of success in Agile?**

There are many indicators that indicate the progression towards success in Agile. Such as whether you are doing the task right, doing it fast, or doing it on time. However, the best measure of success for the Agile teams is customer satisfaction. The means to meet all of users' needs with quality code in a timely manner.

**18.) If the daily stand-up meeting allows you to assess the effectiveness of the sprint and the Sprint demo allows you to assess the effectiveness of the product, what are you assessing in the Sprint retrospective?**

The sprint retrospective allows the group to assess the workflow's effectiveness in a way that emphasizes trust and teamwork. The main purpose is to address key issues about the team without being too confrontational. This kind of discussion can help the team to understand what to keep and what needs to adapt for greater improvement.

**19.) How is stakeholder feedback collected in the Sprint review, and what is done with that feedback?**

The current progress made by the team is demonstrated in a sprint review. The demonstration goes over the software developed and user stories tackled. Any feedback for improvement by stakeholders should be recorded and incorporated afterwards as potential future requirements. The product owner determines which of these items will be addressed in future sprints.

**Chapter 9**

**6.) What is an example of a product differentiator for frozen pizza?**

An example of a product differentiator could be its unique toppings or flavor combinations that set it apart from other brands.

**8.) Should developers participate in focus groups? Why or why not?**

Developers can participate in focus groups, but it may not always be necessary or efficient. Focus groups are typically used to gather feedback from potential users or customers about a product or service. While developers may have valuable insights to contribute, their participation can also influence the group's opinion and skew results.

**9.) When Agile development efforts are committed to the marketplace through marketing and sales, how can the development team maintain flexibility?**

The development team should stay closely aligned with the marketing and sales team throughout the development process. Regular communication and collaboration can help ensure that the product meets market needs and stays responsive to changing market conditions.

**12.) How does a working agreement influence teamwork?**

A working agreement helps to create a shared understanding and sense of accountability among team members, leading to better teamwork and more successful outcomes.

**14.) What are some examples of silos in the marketing organization?**

Silos in marketing organizations can manifest in a number of ways, such as separate teams for social media, email marketing, and advertising. These silos can lead to communication breakdowns, duplicative efforts and missed opportunities.

**17.) Are there any departments or institutions that would not benefit from Agile? Please explain your answer.**

Agile methodologies can be beneficial for any department or institution that values collaboration, flexibility, and continuous improvement. While some departments may have unique challenges or require custom adaptations to the methodology, Agile can generally be applied across a variety of contexts and industries. For example, highly regulated industries such as healthcare or finance may have strict compliance requirements that need to be met which can make it more challenging to implement agile methodologies.

**19.) Since the Agile principles are easy to understand, does that mean they are easy to implement? Please explain your answer.**

While Agile principles are relatively straightforward, implementing them effectively can be challenging. Successful Agile development requires a shift in mindset, culture and processes. Team members must be committed to regular communication, transparency, and continuous improvement. Additionally, Agile methodologies require a level of discipline and rigor to ensure that the team stays on track and delivers value to the customer.