Exam ~~Questions~~ Answers

Here are the answers to this chapter's practice exam questions. How many did you get right? If you got one wrong, that's OK—it's worth taking the time to flip back and re-read the relevant part of the chapter so that you understand what's going on.

1. Answer: D

Did this situation sound negative, like something was going drastically wrong? If it did, you may want to think about your own mindset! This was actually a pretty accurate description of a very successful agile project that uses an iterative methodology. It only sounds like the project is running into problems if you approach it with a mindset that considers change and iteration to be a mistake rather than a healthy activity. If you see the project this way, then you'll be tempted to "blame" the team for not understanding the requirements, or the users for not knowing what they want, or the process for not having adequate controls to prevent and manage changes. Agile teams don't think about things like that. They know that the best way to figure out what the users need is to deliver working software early and frequently.

2. Answer: C

Project managers are very important, but there's no specific role in Scrum called "project manager." Scrum has three roles: Scrum Master, Product Owner, and Team Member. The project manager will fill one of those roles on a project that uses Scrum, but will often still have the "Project Manager" job title.

When your team follows an agile methodology that has specific roles, the role that you fill doesn't always match the title on your business card, especially when your team is just starting to adopt the methodology.

3. Answer: B

It's true that agile teams value customer collaboration, believe face-to-face conversation is the most effective method of conveying information, and place the highest priority on delivering software. However, the user took the time to write the specification, and the information in it could be very helpful in either writing code or having a face-to-face conversation.

When someone takes the time to write down information they think is important, it's very *UN*-collaborative to ignore it.

4. Answer: D

When agile teams talk about working software, they mean software that they consider "done" and ready to demonstrate to the users. But there's no guarantee that it fulfills the users' needs or that it meets the specific requirements in a specification. In fact, the most effective way to build software that genuinely helps users is to deliver working software frequently. The reason is because the early versions of working software typically **don't fully meet the users' needs**, and the only way for everyone to figure that out is to get it into the hands of the users so they can give feedback about it.

This is why agile teams value early and continuous delivery of working software.

Exam Questions Answers

5. Answer: C

The Agile Manifesto contains the core values shared by effective agile teams. It doesn't define a "best" way to build software or a set of rules that all teams should follow, because people on agile teams know that there's no "one-size-fits-all" approach that works for all teams.

6. Answer: B

Scrum teams work in sprints, typically (but not always) 30 days long. They plan the next 30 days of work (assuming the length is 30 days) at the start of the sprint. At the end of the sprint, they demonstrate working software to the users, and also hold a retrospective to review what went well and find ways to improve.

7. Answer: C

The project is suffering because the team is having trouble collaborating with their customer. In this case, the network engineers are the customer, because they're the ones who will be using the software. This is a situation where it would be easy to take a contract negotiation approach, laying out specific terms and documents to describe what will be built so that software development work can begin. But it's more effective to genuinely collaborate with them and work together to discover the best technical solution.

8. Answer: C

A waterfall project is divided into phases, typically starting with requirements and design phases. Many waterfall teams will begin "pre-work" on code once the requirements and design have reached a stable point, even if they're not yet complete. However, this is definitely not the same thing as iteration, because the team doesn't change the plan based on what they learned building and demonstrating working software.

9. Answer: B

Agile projects are built around motivated team members. Keith is taking actions that undermine the whole team's motivation by undercutting a team member who's taking a good risk and genuinely trying to make the project better.

Exam ~~Questions~~ Answers

10. Answer: B

Flip back and reread the first principle of agile: "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software." The reason this is the highest priority is because agile teams are focused first and foremost on delivering software that's valuable. All of the other things we do on projects—planning, design, testing, meetings, discussion, documentation—are really, really important, but it's all in service of delivering that valuable software to our customers.

11. Answer: B

While some teams treat the Daily Standup as a status meeting where each team member gives an update to a boss or project manager, that's not really its purpose. It works best when everyone listens to each other, and uses it to plan the project together as a team.

12. Answer: A

Agile teams value simplicity, because simple designs and code are much easier to work with, maintain, and change than complex ones. Simplicity is often called "the art of maximizing the work not done" because—and this is especially true of software—the most effective way to keep something simple is often to simply do less.

13. Answer: D

The reason that the team isn't paying attention during the daily standup is because they don't really care about it or buy into it as an effective tool, and mainly want it to end as quickly as possible so they can get back to their "real" jobs. When teams have this mindset, it's likely that they will eventually stop attending the meeting altogether, and the agile adoption is much less likely to be successful. The daily standup practice will be more effective if the team understands how it helps each of them, both individually and as a team. That mindset shift can only be accomplished through open and honest discussion about what's working and what isn't. That's why working with the team on changing their mindset is the best approach to this situation.

14. Answer: B

It certainly makes sense to read and understand the specification. But the most effective way to truly ascertain whether or not the team really understands what she intended is to deliver working software to her, so that she can see how the requirements she documented were interpreted and work with the team to determine what works well and what needs to change.

Exam Questions Answers

15. Answer: A

There are a lot of great reasons that agile teams welcome changing requirements. When customers are encouraged to change their minds (rather than discouraged from it), they give better information to the team, and that leads to better software. And even when people keep their mouths shut about changes, they almost always eventually get exposed in the end, so when the team gets them early it gives them more time to respond—and the earlier the changes arise, the less code has to be reworked.

However, changes are never an excuse for poor planning or missed deadlines. Effective agile teams generally have an agreement with their users: the teams welcome changing requirements from users, customers, and managers, and in return they aren't blamed for the time it takes to respond to those changes, because everyone recognizes it's still the fastest and most effective way to build software. So nobody really sees welcoming changing requirements as giving the team a way to explain a missed deadline, because the deadlines should already be adjusted to account for the changes.

16. Answer: A

Working software is delivered frequently so that the team can get frequent feedback and make changes early. That's why working software should never be assumed to contain the final version of any requirement. That's why it's "working" software, not "finished" software.

17. Answer: A

Iterations are timeboxed, which means that the deadline is fixed and the scope varies to fit it. The team starts each iteration with a planning meeting to decide what work will be accomplished. But if it turns out that they didn't get the plan right and work takes longer than expected, then any work that didn't get done is returned to the backlog and reprioritized (and often ends up in the next iteration).