**Questions about Software Lifecycle and Team Management:**

**What is agility?**

*Agility is the ability to be quick and graceful. You might have agility on the basketball court or in the courtroom, or even with your gaming remote. The noun agility can be used for both mental and physical skills in speed and grace. Your mental agility might allow you to follow both conversations at once.*

*Agile Software Development is a lightweight software engineering framework that promotes iterative development throughout the life-cycle of the project, close collaboration between the development team and business side, constant communication, and tightly-knit teams. Also known as agile development.*

**How would you deal with Legacy Code?**

*Eight Strategies For Tackling Legacy Code You Didn't Write*

*Start Small.*

*Don't Judge.*

*Talk To Your Peers.*

*It's All In The (Variable) Name.*

*Be Conventional.*

*When In Doubt, Test Everything. Rinse And Repeat.*

*Learn Different Programming Languages.*

*8. Make It Personal.*

**Say I'm your Project Manager, and I'm no expert in programming. Would you try explaining me what Legacy Code is and why should I care about code quality?**

**I'm the CEO of your company. Explain to me Kanban and convince me to invest in it.**

**What is the biggest difference between Agile and Waterfall?**

*Here are three main reasons why code quality is important: ... Lower technical debt: Good quality code can speed up long-term software development since it can be reused and developers don't have to spend that much time fixing old bugs and polishing code. It also makes it easier for new project members to join the project.*

*Waterfall is a structured software development methodology, and often times can be quite rigid, whereas the Agile methodology is known for its flexibility. ... One of the major differences between Agile and Waterfall development methodology is their individual approach towards quality and testing.*

**Being a team manager, how would you deal with the problem of having too many meetings?**

**How would you manage a very late project?**

*Scenario: The project is forecast to run late. ...*

*Solution 1: Consult with your stakeholders to agree on a suitable extension of time. ...*

*Solution 2: Prioritise project elements, reduce project scope or agree on staged delivery. ...*

*Solution 3: Bring on additional resources and/or extend working hours (spend more money)*

**"Individuals and interactions over processes and tools" and "Customer collaboration over contract negotiation" comprise half of the values of the Agile Manifesto.**

**Discuss**

**Tell me what decisions would you take if you could be the CTO of your Company.**

*1. Individuals and Interactions Over Processes and Tools*

*2. Working Software Over Comprehensive Documentation*

*3. Customer Collaboration Over Contract Negotiation*

*4. Responding to Change Over Following a Plan*

*5. Customer satisfaction through early and continuous software delivery –*

[*https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto*](https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto)

**Organize a development team using flexible schedules (that is, no imposed working hours) and "Take as you need" vacation policy**

**How would you manage a very high turn over and convince developers not to leave the team, without increasing compensation? What could a Company improve to make them stay?**

*Lack of clarity about expectations*

*Lack of clarity about earning potential*

*Lack of feedback about performance*

*Failure to hold scheduled meetings*

*Failure to provide a framework within which the employee perceives they can succeed''*

**What are the top 3 qualities you look for in colleagues, beyond their code?**

**What are the top 3 things you wish non-technical people knew about code?**

**Imagine your company gives you 1 month and some budget to improve your and your colleagues' daily life. What would you do?**