**Intro to Agile**

|  |  |
| --- | --- |
| **Q. 1** | **What is Agile Software Development & why we use Agile ?** |
| Ans. | Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams  **Why we use Agile ?**   * Improved return on investment (RIO) * Early detection and cancellation of failing products * Higher quality software * Improved control of a project * Reduced dependence on individuals and increased flexibility |
| **Q. 2** | **What are the core values of the manifesto and what are the Agile Principles ?** |
| Ans. | **The core values of Agile Manifesto:**   1. Individuals and interaction take precedence over processes and tools. 2. Working software takes precedence over comprehensive documentation. 3. Customer collaboration takes precedence over contract negotiation. 4. Responding to change takes precedence over following a plan.   **The Principles of Agile Methods**   1. Our highest priority is to satisfy the customer through the early and continuous delivery of software. 2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. 3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. 4. Business people and developers must work together daily throughout the project. 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. 7. Working software is the primary measure of progress. 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. 9. Continuous attention to technical excellence and good design enhances agility. 10. Simplicity--the art of maximizing the amount of work not done--is essential. 11. The best architectures, requirements, and designs emerge from self-organizing teams. 12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. |
| **Q. 3** | **What is Scrum ? What are the Core Practices & Artifacts of Scrum ?** |
| Ans. | **Scrum is a:**   * Agile way of project management * A team based collaborative approach * Iterative & incremental development * Always focus to deliver “Business Value”   **Core Practices & Artifacts :**   * User Story * Sprint * Release Planning Meeting * Sprint Planning Meeting * Daily Scrum Meeting (Daily Stand up) * Sprint Review Meeting * Retrospective * Product Backlog * Sprint Backlog * Burn-Down Chart * Velocity * Impediment Backlog |
| **Q. 4** | **What is a user story & give the template with example ?** |
| Ans. | **User Story :**   * Expressing requirements in agile processes is called as a User Story * User Story is an effective approach on all time-constrained projects * It contains a name, a brief narrative, and acceptance criteria and conditions for the story to be complete * The advantage of user stories is that they focus on exactly what the user needs/wants without going into the details on how to achieve it   **User Story Template :**   * Role: He is the owner of the User Story. This is often a user. By using specific users like, Administrator, Student, Logged in User, Unauthenticated User etc. it’s easier to understand and sets the user story into context * Feature: It describes what user wants to do on the system. If it is a mandatory action it can be prefixed by "must". Otherwise "want" is used. * Achievement: It describes the outcome resulted from actions.     **Example :**  As an <applicant> I want to <see all open positions in the organization> so that <I can apply for the suitable position> |
| **Q. 5** | **What is Extreme Programming and mention its principles ?** |
| Ans. | **Extreme Programming:**  Extreme Programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted.  **Principles of Extreme Programming:**   * The Planning Game * Small Releases * Metaphor * Simple Design * Testing * Refactoring * Pair Programming * Collective Ownership * Continuous Integration * 40-hour Week * On-site Customer * Coding Standards |
| **Q. 6** | **What is Lean Software Development and mention its principles ?** |
| Ans. | **Lean Software Development :**  Lean software development is a management philosophy that focuses on throughput. Lean software development doesn't focus on particular components of the value-stream like code-construction or QA, but on whether the components of the chain are working as efficiently as possible so as to generate as much value as possible to the customer.  **Principles of Lean Software Development:**   * Eliminate waste * Amplify Learning * Decide as late as possible * Deliver as fast as possible * Empower the team * Build integrity * See the whole |
| **Q. 7** | **What is Agile Testing and where does tester fit in the Agile team?** |
| Ans. | **Agile Testing:**   * Agile testing is a software testing practice that follows the principles of agile software development. * Agile testing involves all members of a cross-functional agile team, with special expertise contributed by testers, to ensure delivering the business value desired by the customer at frequent intervals, working at a sustainable pace.   **Role of a tester in Agile team:**   * Sometimes some agile teams don’t have any member in particular who define themselves as testers. * However, they all need someone who can assist the customer team in writing functionalities facing test for user stories of each iteration, ensure that the tests pass, and make sure that adequate regression tests are automated. * Even under the situation when the team does have testers, the whole agile team would be responsible for the testing of iteration user stories. |
| **Q. 8** | **What is Iteration Zero and what are the factors which influence it ?** |
| Ans. | **Iteration Zero:**   * Iteration Zero is a focused set of activities that a team does to get ready to begin a series of product development iterations   **Factors influencing Iteration Zero:**   * The maturity of the team * Whether stakeholders have worked on an Agile project before * The quality of the user stories or understanding of the requirements that are being developed * The environments being used * Whether the environments are already setup * The level of experience the organization has with agile methods |
| **Q. 9** | **What is Agile Testing Quadrant ?** |
| Ans. | **Agile Testing Quadrants :**  **Quadrant 1 :** Automated Testing  **Quadrant 2 :** Automated & Manual Testing  **Quadrant 3 :** Manual Testing  **Quadrant 4 :** Testing with tools. |
| **Q. 10** | **Explain Test Planning in Agile Testing** |
| Ans. | * On Agile projects the teams don’t depend upon heavy documentation about what testers need to do * Testers works with the agile team hand in hand so that the testing efforts are visible to all in the form of testing cards * In release planning the agile team defines the purpose of the release, scope and assumptions * The test plan should contain information about testing issues that are specific to this release or project * It should contain risk analysis and assumptions identified * The test plan should outline the critical success factors that your customers has identified * Keep only that information which people need to know about testing and remove any extraneous information * The main advantage of test planning is planning itself * It allows you to consider and address issues such as test data requirements, infrastructure * Test planning is a risk mitigation strategy |