Agile testing

Agile testing is a software testing process that follows the principles of agile software development.Agile testing aligns with iterative development methodology in which requirements develop gradually from customers and testing teams. The development is aligned with customer requirements.

Agile testing is a continuous process rather than being sequential. The testing begins at the start of the project and there is ongoing integration between testing and development. The common objective of agile development and testing is to achieve a high product quality.

Principles of Agile Testing:

- Testing is continuous
- Continuous feedback
- Tests performed by the whole team
- Decrease time of feedback response
- Simplified & clean code
- Less documentation
- Test Driven

There are various agile testing methods as follows:

- Behavior Driven Development (BDD)
- Acceptance Test Driven Development (ATDD)
- Exploratory Testing

Advantages:

- As testing begins at the start of the project, errors can be fixed in the middle of the project.
- There is very less documentation required for agile testing.
- User acceptance is performed at the end of every sprint.
- Face-to-face conversation is the best form of communication.
- Close, daily cooperation between business people and developers.

Disadvantages:

- Not so good for large projects .

- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.

The agile testing lifecycle includes the following 5 phases:

- Impact assessment
- Agile Testing Planning
- Release Readiness
- Daily Scrums
- Test Agility Review

When to use agile testing:

- When new changes are needed to be implemented.
- Unlike the waterfall model in agile model very limited planning is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way.

Agile Quadrants