Agile Testing

Vu Lam

Agile Process

- Scrum
- Crystal Methodologies
- DSDM (Dynamic Software Development Method)
- Feature driven development (FDD)
- Lean software development
- Extreme Programming (XP)

What is Agile Testing?

- AGILE TESTING is a testing practice that follows the rules and principles of agile software development.
- Unlike the Waterfall method, Agile Testing can begin at the start of the project with continuous integration between development and testing.
- Agile Testing methodology is not sequential (in the sense it's executed only after coding phase) but continuous.

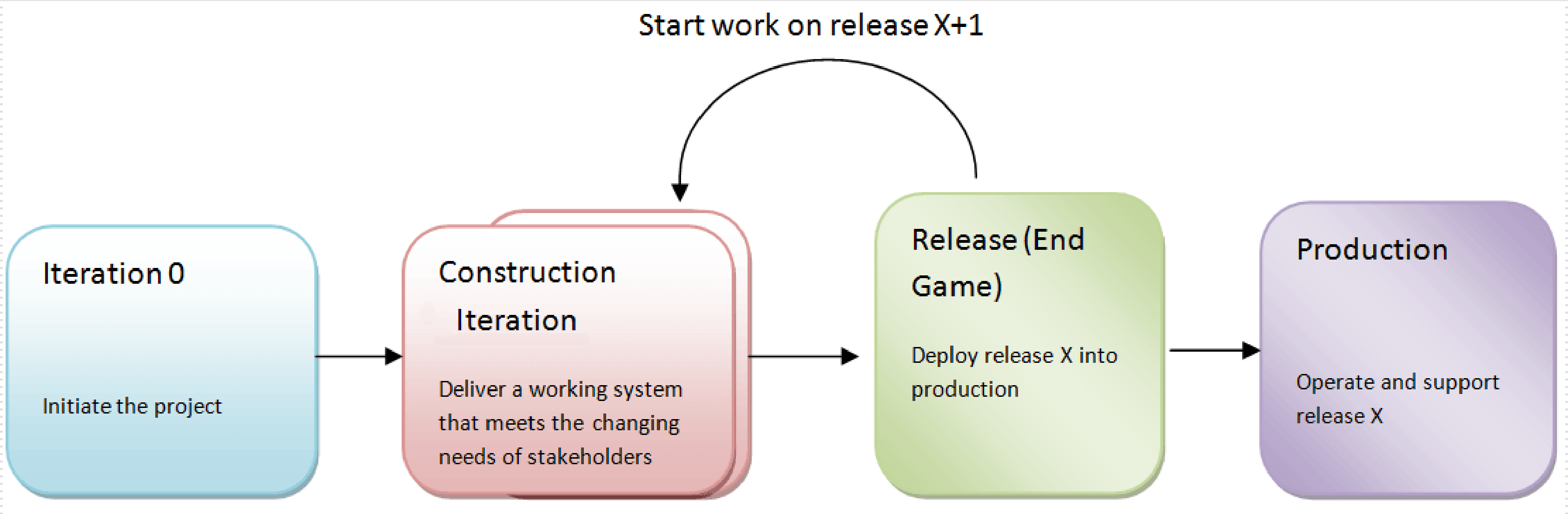
Agile Test Plan

- Agile test plan includes types of testing done in that iteration like:
  - test data requirements,
  - infrastructure,
  - test environments,
  - test results.
- Unlike the waterfall model, in an agile model, a test plan is written and updated for every release

Agile test plans

- Testing Scope
- New functionalities which are being tested
- Level or Types of testing based on the features complexity
- Load and Performance Testing
- Infrastructure Consideration
- Mitigation or Risks Plan
- Resourcing
- Deliverables and Milestones

Agile Testing Strategies


Start work on release X+1
Iteration 0
Initiate the project
Construction Iteration
Deliver a working system that meets the changing needs of stakeholders
Release (End Game)
Deploy release X into production
Production
Operate and support release X

Agile Testing Strategy

(a) Iteration 0

- During the first stage or iteration 0, you perform initial setup tasks. It includes identifying people for testing, installing testing tools, scheduling resources (usability testing lab), etc. The following steps are set to achieve in Iteration 0:
  - Establishing a business case for the project
  - Establish the boundary conditions and the project scope
  - Outline the key requirements and use cases that will drive the design trade-offs
  - Outline one or more candidate architectures
  - Identifying the risk
  - Cost estimation and prepare a preliminary project

(b) Construction Iterations

- A set of iterations to build an increment of the solution.
- Construction iteration is classified into two, confirmatory testing and investigative testing.
  - Confirmatory testing concentrates on verifying that the system fulfills the intent of the stakeholders as described to the team to date, and is performed by the team
  - Investigative testing, tester determines the potential problems in the form of defect stories. Investigative testing deals with common issues like integration testing, load/stress testing, and security testing.

Confirmatory testing

- There are two aspects of confirmatory testing: developer testing and agile acceptance testing.
  - Both of them are automated to enable continuous regression testing throughout the lifecycle.
  - **Agile acceptance** testing is a combination of traditional functional testing and traditional acceptance testing as the development team, and stakeholders are doing it together
  - **Developer testing** is a mix of traditional unit testing and traditional service integration testing. Developer testing verifies both the application code and the database schema.

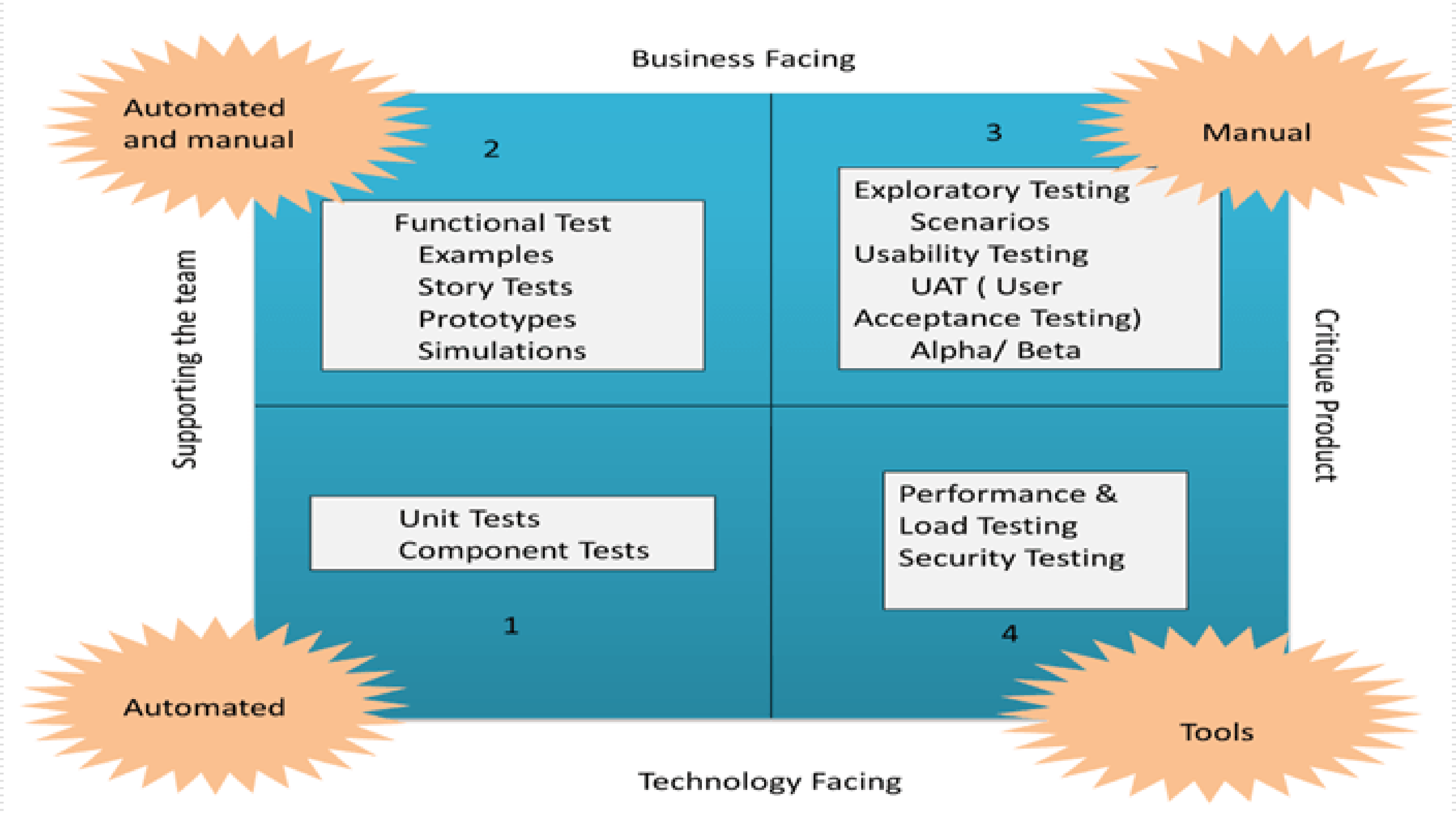
(c) Release End Game Or Transition Phase

- The goal of "Release, End Game" is to deploy your system successfully into production.
  - The activities include in this phase are training of end users, support people and operational people. Also, it includes marketing of the product release, back-up & restoration, finalization of system and user documentation.
- Final agile methodology testing stage includes full system testing and acceptance testing

(d) Production

- After the release stage, the product will move to the production stage.

The Agile Testing Quadrants



Challenges with agile testing

- Chances of error are more in agile, as documentation is given less priority, eventually puts more pressure on QA team
- New features are introduced quickly, which reduces the available time for test teams to identify whether the latest features are according to the requirement and does it truly address the business suits
- Testers are often required to play a semi-developer role
- Test execution cycles are highly compressed
- Very less time to prepare test plan
- For regression testing, they will have minimal timing
- Change in their role from being a gate-keeper of quality to being a partner in Quality
- Requirement changes and updates are inherent in an agile method, becoming the biggest challenge for QA

Risk of Automation in Agile Process

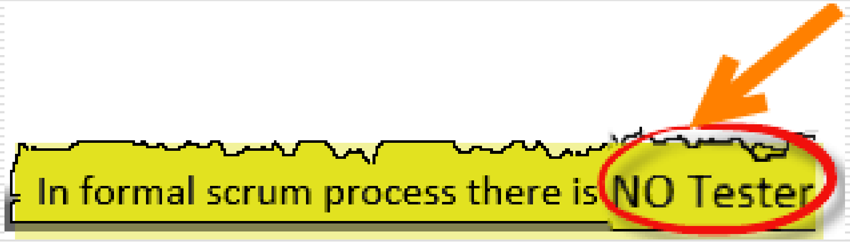
- Automated UI provides a high level of confidence, but they are slow to execute, fragile to maintain and expensive to build.
- Unreliable tests are a major concern in automated testing. Fixing failing tests and resolving issues related to brittle tests should be a top priority in order to avoid false positives
- If the automated test are initiated manually rather than through CI (Continuous Integration) then there is a risk that they are not regularly running and therefore may cause failing of tests.
- Automated tests are not a replacement for an exploratory manual testing. To obtain the expected quality of the product, a mixture of testing types and levels is required
- Many commercially available automation tools provide simple features like automating the capture and replay of manual test cases. Such tool encourages testing through the UI and leads to an inherently brittle and difficult to maintain tests. Also, storing test cases outside the version control system creates unnecessary complexity

Risk of Automation in Agile Process

- In order to save time, much times the automation test plan is poorly planned or unplanned which results in the test fail
- A test set up and tear down procedures are usually missed out during test automation, while Performing manual testing, a test set up and tear down procedures sounds seamless
- Productivity metrics such as a number of test cases created or executed per day can be terribly misleading, and could lead to making a large investment in running useless tests
- Members of the agile automation team must be effective consultants: approachable, cooperative, and resourceful, or this system will quickly fail
- Automation may propose and deliver testing solutions that require too much ongoing maintenance relative to the value provided
- Automated testing may lack the expertise to conceive and deliver effective solutions
- Automated testing may be so successful that they run out of important problems to solve, and thus turn to unimportant problems.

Role of Tester in Scrum

- There is no active role of Tester in the Scrum Process.
- Usually, testing is carried out by a developer with Unit Test.
- While product owner is also frequently involved in the testing process during each sprint.
- Some Scrum projects do have dedicated test teams depending on the nature & complexity of the project.



In formal scrum process there is NO Tester

Sprint Planning

- In sprint planning, a tester should pick a user-story from the product backlog that should be tested.
- As a tester, he/she should decide how many hours (Effort Estimation) it should take to finish testing for each of selected user stories.
- As a tester, he/she must know what sprint goals are.
- As a tester, contribute to the prioritizing process

Sprint

- Support developers in unit testing
- Test user-story when completed. Test execution is performed in a lab where both tester and developer work hand in hand. Defect are logged in Defect Management tool which are tracked on a daily basis.
- As a tester, he/she attends all daily stand up meeting to speak up
- As a tester, he/ she can bring any backlog item that cannot be completed in the current sprint and put to the next sprint
- Tester is responsible for developing automation scripts. He schedules automation testing with Continuous Integration (CI) system.

Sprint

- Review CI automation results and send Reports to the stakeholders
- Executing non-functional testing for approved user stories
- Coordinate with customer and product owner to define acceptance criteria for Acceptance Tests
- At the end of the sprint, the tester also does acceptance testing(UAT) in some case and confirms testing completeness for the current sprint

Sprint Retrospective

- As a tester, you will figure out what went wrong and what went right in the current sprint
- As a tester, you identify lesson learned and best practices