ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Strategic Practices of Testing Software Applications in Agile Methodology.

[1] Mageusha.U,[2] Dr. Sudhamani
[1] Research Scholar,Rayalaseema University
[2] Research Supervisor, Principal, Vivekananda Degree College for women.
Mahalakshmi Layout, Bangalore.

Abstract: A Strategic practices of testing software applications must concentrate on low level testing practices that are important to verify that unit level source code segment is correctly implemented as well as other tests like high level such as major work flows of the requirement that validate major application functionality against user requirements. A strategy need to help the testing team and a set of milestones for the test management. The strategic practices need to be implemented at a time when dead-line pressure begins to rise, progress must be predictable and measurable and problems must find out in advance at the earliest.

Keywords: Testing Practices, Application functionality, Testing-team, dead-line, Test Management.

INTRODUCTION

The Test strategic practices define on a high level the various test phases and releases of the programme and the testing process and methodology with entry and exit criteria for phases. It furthermore defines the ways of working and the scope elements for delivering integrated product testing within testing team. The practices also highlight testing areas of unique complexity and describes the tools and services that will support the functional test phases defined as well as the risks and mitigating actions for testing.

The main purpose of this paper is to detail the test Strategy and practices to be employed for all the testing stage of the releases.

The Test Strategy outlines the guidelines and definition, which is used as the basis for the Test Approach. Test Approach serves as an input to subsequent Test Approaches developed for specific phases of testing by addressing all major aspects of the Programme Test Phase that affect the success of testing and does not cover all the aspects of these respective approaches. It is therefore importance that these deliverables are read alongside the Test Approach to give an overall view of the testing performed by test team in the various phases of testing and across all aspects of the programme.

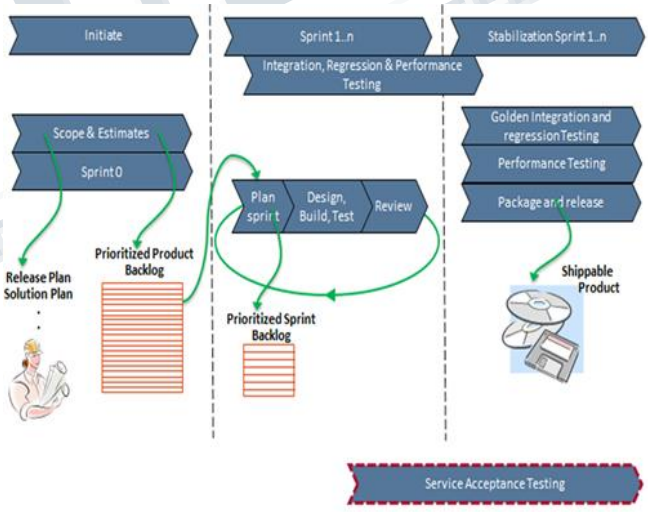

Fig 1.0 Strategic practices of testing software applications:

Risks and Assumptions:
Testing team should have assumptions before they start actual software application testing in agile methodology, those assumptions are

- Component testing will be owned and performed by the Scrum team.
- User Acceptance Testing will be owned and performed by the product management team.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

- Detailed specifications of test requirements are documented in the user stories.
- New and existing hardware is set-up and functional before start of testing.
- Issue and defect resolution support is covered by Scrum.
- Any scope changes to the release will be communicated to the test team. Before the scope change is finally agreed the test team has evaluated the impact on the test planning for the release and confirmed that the changed scope can be included into the plan.
- Any impact to the testing scope, timeline and overall release will be evaluated and confirmed by the test team.
- Testers will be available and trained in the application and tools prior to the commencement of test preparation.
- The test environments will be available to the testing team in adequate time for shakedown and test execution.
- Service Acceptance Test scope will be integrated into the Product Engineering Tests and executed by the Testing Factory team.

Testing Phases:

1. Unit / Component Test

Scope/Objective

The objective of Unit Testing is

- To verify that each application component is properly implemented and technically stable
- To validate that a module reflects the requirements and specifications documented in the User Stories

Approach

All new and modified modules being implemented as part of this release will be unit tested which includes:

- Method sequence and Interactions.
- API testing covering verify functionality and expose failures.
- Common parameter values exploring boundary conditions.
- Modular requirements are captured in the use es and technical designs.
- Restart/Recovery for batch processes.
- Basic data field validations.
- Logical flow checks.
- Min/ max conditions
- Conditions, Loops, Control Breaks, etc.
- Error and exception handling

2.System/ FunctionalTest

Scope/Objective

The purpose of the assembly test phase is to validate that data flows through all connection points of the system/systems and middleware per the requirements and design documents.

Capturing Response time is based on the product line under test, which includes

- CP Time is used to measure the response time for the steps documented in the click list.
- Tools: CP Time is both for Mobile and Desktop applications.
- Performance Counter: Using Microsoft Visual Studio Ultimate & Microsoft Performance Monitor.
- DB Table/API features of workers: KPIs for Integration/Replication.
- Click Lists: using stop watch & excel.
- By including the time measurements into the source code.

Approach

Assembly test is performed to test the Conditions specific to the Module, exception conditions, Negative /Error conditions. When applicable, each module will be tested and validated against the following criteria:

- Functional Tests
- Error/Exception Tests
- Conditions specific to a Module

3.Project / System Integration Test

Scope/Objective

The purpose of the Product Integration Test phase is to confirm that the application changes meet the documented functional, technical and quality requirements of the release. During integration test data quality and correctness will be validated in and across all the modules and systems within the scope of the release like:

- Detailed data validation
- Real life scenarios covered

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

- End-to-End testing of the entire system.
- Detailed data validation.
- Real life scenarios covered
- End-to-End testing of the entire system and or systems

Approach

Find as many defects as possible as early as possible by executing as many product integration test scripts as possible. It is not essential that the entire cycle be completed if major defects are discovered. Ideally the whole functionality and all affected areas will be covered in width.

4.Regression test

Scope/Objective

Regression Test is performed to validate that existing functionality is not negatively impacted by the introduction of the new application/functionality being implemented for the release.

The tests conducted in this phase are either

Approach

- The regression testing process involves testing existing systems being changed due to approved change requests or new releases
- Regression Testing ensures that all components working previously are not adversely impacted by the change in the configuration and in the application.
- For later releases the regression test ensures that functionality which was working before the go-live of the respective release stills works.
- Automation Testing helps in reducing the time taken to execute the regression testing scripts

5.User Acceptance Test

Scope/Objective

The purpose of the user acceptance test (UAT) phase is to confirm that the application changes meet the functional, technical and quality requirements of the business teams from an end user perspective.

Approach

All UAT scripts (defined by scenarios in requirement documents and demo click lists) will be executed and re-executed until the scripts have been completed successfully.

6. Set-Up Test

Scope/Objective

The purposes of the Operational Readiness Test (Setup Test)is to confirm that

- The different setups can be installed successfully on the corresponding environments
- (e.g. operation systems, database systems, etc.).
- Installed environments work together (e.g. deployment, replication).
- Test environment is prepared for a full regression test.

Approach

A standard change management procedure/process will be followed for transferring applications/ scripts/ procedures from Development to Test/QA and into the Production environment. Guidelines and Instructions are available for Set-up, System Requirements, 3rd Party Installation, Compatibility documents available at the below URL are followed:

7.Migration Test

Scope/Objective

The purpose of the migration test phase is to confirm that:

- Possible to migrate existing project environments and production environments to the latest version.
- Migrated system meets the documented functional, technical and quality requirements of the release.

Approach

The test approach for database migration testing consists of the following activities:

- Design the validation tests
- Run your validation tests:
- Report the bugs and call the exterminator

8.Service Acceptance Test

Scope/Objective

SAT checks readiness of a new release before general availability (e.g. before the first project)

- System Setups
- Stability
- Functionality
- Extendability (customizing)

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

- Upgradeability (migration)
- Availability of documentation and training material
- Usability
- Completeness

Approach

- Final Integration Test for each release starts with
- Setting up a new system
- Upgrading a reference system

8.Performance Test

Scope/Objective

During performance tests, response times and data flows will be validated according to the requested performance specifications within the requirements. The core product is tested in a dedicated environment for performance tests. Strategic Processes of Software Testing:

- Test Coverage
- Test Plan & design
- Equivalence Partitioning
- Boundary Value Analysis
- Test Preparation and Execution
- Defect Life Cycle
- Root Cause Analysis
- Reporting
- Environments

CONCLUSION

Strategic practices to be considered for identifying appropriate test approach for the software (Project or Product) testing:

The following points should be identified when defining the strategy or approach:

- Risks and Assumptions related to targeted testing practices.
- Skilled and Experienced resources
- The nature of the software environment and the domain.
- Proposed techniques, practices and methods.
- The nature of the product and the business,
- The objective of the testing team.

REFERENCES:

[1] B. Beizer, Software Testing Techniques, Second Edition. Published by dreamtech, New Delhi, pp. 135-143, 2008

[2] Parashar,A.Kalia,R.Bhatia,"ChangeImpactAnalysis:ATool for EffectiveRegressionTesting,".

[3] Mathur, AP (2008) "Foundations of software testing". Dorling, Kindersley (India), Pearson Education in South Asia, Delhi, India.

[4] Z. Li, M. Harman, and R. M. Hierons, "Search algorithms for regression test case prioritization," IEEE Transactions on Software Engineering, vol.33, no. 4.

[5] Testing Strategies and Tactics for Mobile Applications [Online]. Available: http:// www. keynote. com/mobile-app-testing.html.