InnerThink

“Best” Practices for Test Management

Contents

Objective: To understand what Test Management is, why it is important, what is a Best Practice, what it constitutes, and then begin to apply Test Management practices to what we deliver as part of our DNA. Note: This document draws upon multiple sources, reflecting existing industry insights, to pull together an end-to-end view.

InnerThink

Best Practice Defined

Definition

1. A method or technique that has consistently shown results superior to those achieved with other means, and that is used as a benchmark. See also best in class and leading practice. **Source:** BusinessDictionary.com
2. Commercial or professional procedures that are accepted or prescribed as being correct or most effective. **Source:** Oxford Dictionaries.

NOTE:

Testing is not stand-alone. It is dependent on preceding delivery activities (development, configuration) and is subject to the practices used, as well as the baseline requirements and how they are organized and managed. However, testing itself is a separate set of efforts used to validate that what is delivered meets the objectives of the initiative (Scope, Requirements, Quality, Fit-for-use).

InnerThink

What is Test Management?

Overview

Test Management is the practice of organizing and controlling the process of artifacts required for a testing effort, as well as, the testing effort itself.

The goal is to enable teams to plan, develop, execute, and assess all testing activities within the overall delivery (development) effort. This includes coordination of efforts related to the testing effort, tracking of dependencies and relationships among test assets, and most importantly, defining, measuring, and tracking quality goals.

Level of rigor, detail, and scope will vary depending on aspects such as:

- In-house vs. Cloud based service
- Out-of-box vs. customized
- Mobile vs. ERP solutions
- Application vs. Infrastructure
- Technical vs. Functional
- Integrated vs. Non

As well, regardless of Delivery Methodologies used, all initiatives need to incorporate a level of Test Management discipline in order to effectively validate what is delivered.

InnerThink

Elements of Test Management

Test artifact and resource organization; inventory of items to test, along with the various things used to perform the testing. This addresses how teams track dependencies and relationships among test assets. The most common types of test assets that need to be managed are:

1. Test scripts
2. Test data
3. Test software
4. Test hardware

Test planning is the overall set of tasks that address the questions of why, what, where, and when to test. The reason why a given test is created is called a test motivator (for example, a specific requirement must be validated). What should be tested is broken down into many test cases for a project. Where to test is answered by determining and documenting the needed software and hardware configurations. When to test is resolved by tracking iterations (or cycles, or time period) to the testing.

Elements of Test Management

Test authoring is a process of capturing the specific steps required to complete a given test. This addresses the question of how something will be tested. This is where somewhat abstract test cases are developed into more detailed test steps, which in turn will become test scripts (either manual or automated).

Test execution entails running the tests by assembling sequences of test scripts into a suite of tests. This is a continuation of answering the question of how something will be tested (more specifically, how the testing will be conducted).

Test reporting is how the various results of the testing effort are analyzed and communicated. This is used to determine the current status of project testing, as well as the overall level of quality of the application or system.

Industry Practices

Overview

The following practices are highlights from IBM Research Technical Report RC 21457.

Basic Practices; employ:

- Functional Specifications: Describes a procedure indicating options that a service can be run. This serves to ensure that there is alignment between what the Design is and what is then to be delivered. Test cases can then be developed from these artifacts. This can then be done in parallel to the Development phase.
- Reviews & Inspections – pre-emptive prior to formal test cycles, e.g. Design Review
- Formal Entry & Exit Criteria
- Functional Testing with scenario (condition) variations
- Beta's – attain early feedback and identify issues. Increase adoption to larger audiences
- Automated test execution

Industry Practices

Foundational; employ

- User Scenarios that test functionality of the solution reflecting “how” the User will use it rather than focused on ‘Features’
- Usability Testing; end quality focus. How “usable” is the end solution (ease-of-use), what it the users experience; solicit feedback on how the solution is actually used and improve
- Defect tracking / measures and feedback loop into design and delivery
- Requirements for test planning – they are the baseline to test planning

Industry Practices

Incremental; employ

- Align Testers with the Delivery team (Developers, Analysts). Engage thru life-cycle of the project to avoid SME loss, disconnects with context, and delays in decisions. This can expedite test case preparations, execution, and defect resolution.
- Code Coverage; measure the elements of code that have been tested.
- Automatic Environment Generation; setting up / re-configuring environments for testing can be arduous. Leverage Infrastructure as service, or utilities to setup environments, run tests, and record results. HIGH MATURITY ELEMENT
- Test on Demand; As changes are introduced test and delivery in part or whole. Re-purpose tools, cases, and techniques particularly of value add with SaaS solutions, or, where there are iterative deployments of functionality based on a consistent platform or solution
- State task Diagram; illustration of functional view of what is being delivered in order to create comprehensive test cases
- Statistical Testing; determining reliability as opposed to debugging process issues. Test a function and measure the interfailure rates; a good development process should in fact result in an increase of the mean-time between failure and when a defect/bug is fixed.
- Check-In Tests for Code; unless code passes a 'unit' test it does not get promoted into the next build
- Minimize Regression testing; focus on key integration points and dependencies of what a change impacts / influences
- Bug Bounties: Focused effort to identify bugs, not just pass a scenario, determine what can cause things to break, not just validating something does what it is supposed to do.

InnerThink

Industry Practices

Additional Considerations:

- Testing should be considered an element of Quality Assurance, but not the only factor. Testing will unveil challenges with process, design, and development that of themselves need to consider a pro-active plan-do-check mind set.
- Defects will result in the need for a Risk & Change Management discipline
- Employ standard templates to ensure completeness needed to effectively execute a test condition
- Employ Web based tools to support Test Management activities and artifacts

- From Gartner (elements covered in preceding slides):
  - Employ Continuous testing
  - Automate
  - Don't wait for full completion before beginning testing
  - Ensure code/configuration baseline is stable during testing
  - Test early and often
  - Be pragmatic about adding tests to untested legacy code

InnerThink

Keys to Success

Keys to Success, or, Challenges to address for Success

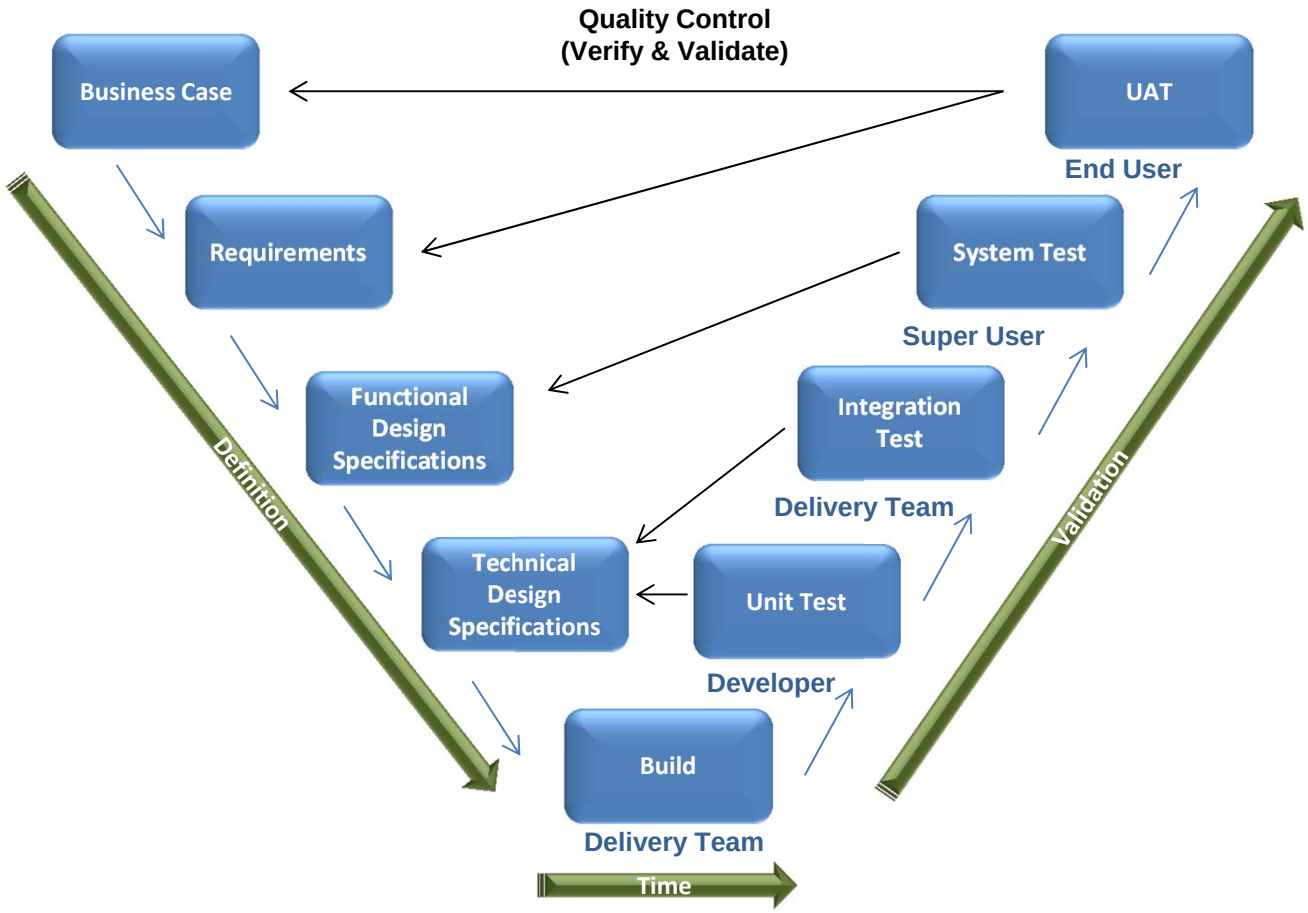
- Have enough resources to test (people, hardware, tools)
- Have the right collaboration framework to share artifacts and hand-off between multiple geographic regions
- Have a clear set of Requirements to validate against (Signed-off)
- Report the right information out, and define the right metrics to manage against
- Test early on, iterate, don't wait for a one-time formal test cycle. Get results early on and minimize opportunity costs and lost time
- Re-purpose test artifacts
- Leverage remote testers, enhance buy-in and use-case variations
- Employ a nimble, flexible, process. However, enforce consistency and standards to comply with regulatory controls
- Employ consistent tool, process, and disciplines
- Have the right 'decision makers' engaged to address testing outcomes
- Have executive support to champion discipline

InnerThink

Baseline Testing Guidance

Embrace Quality Assurance:

Tie back testing to what we set out to do. The V-Model demonstrates the relationships between what testing occurs, at what general stage (be it an iterative, sequential, or agile employed methodology), and against what types of project artifacts. The nature of the project will determine what is needed and when, and to what level of grain.



InnerThink

Appendix A – Other Resources

Benchmark Definition:

Standard, or a set of standards, used as a point of reference for evaluating performance or level of quality. Benchmarks may be drawn from a firm's own experience, from the experience of other firms in the industry, or from legal requirements such as environmental regulations.
Source: BusinessDictionary.com

Testing Definition per TMMi:

The process consisting of all life-cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.

Other Resource References:

- IBM Testing Management
- Best Practices.com
- Software Testing Help
- Rishahsoft
- TMMi
- Deloitte
- PMI

InnerThink

Appendix B - Baseline Testing Guidance

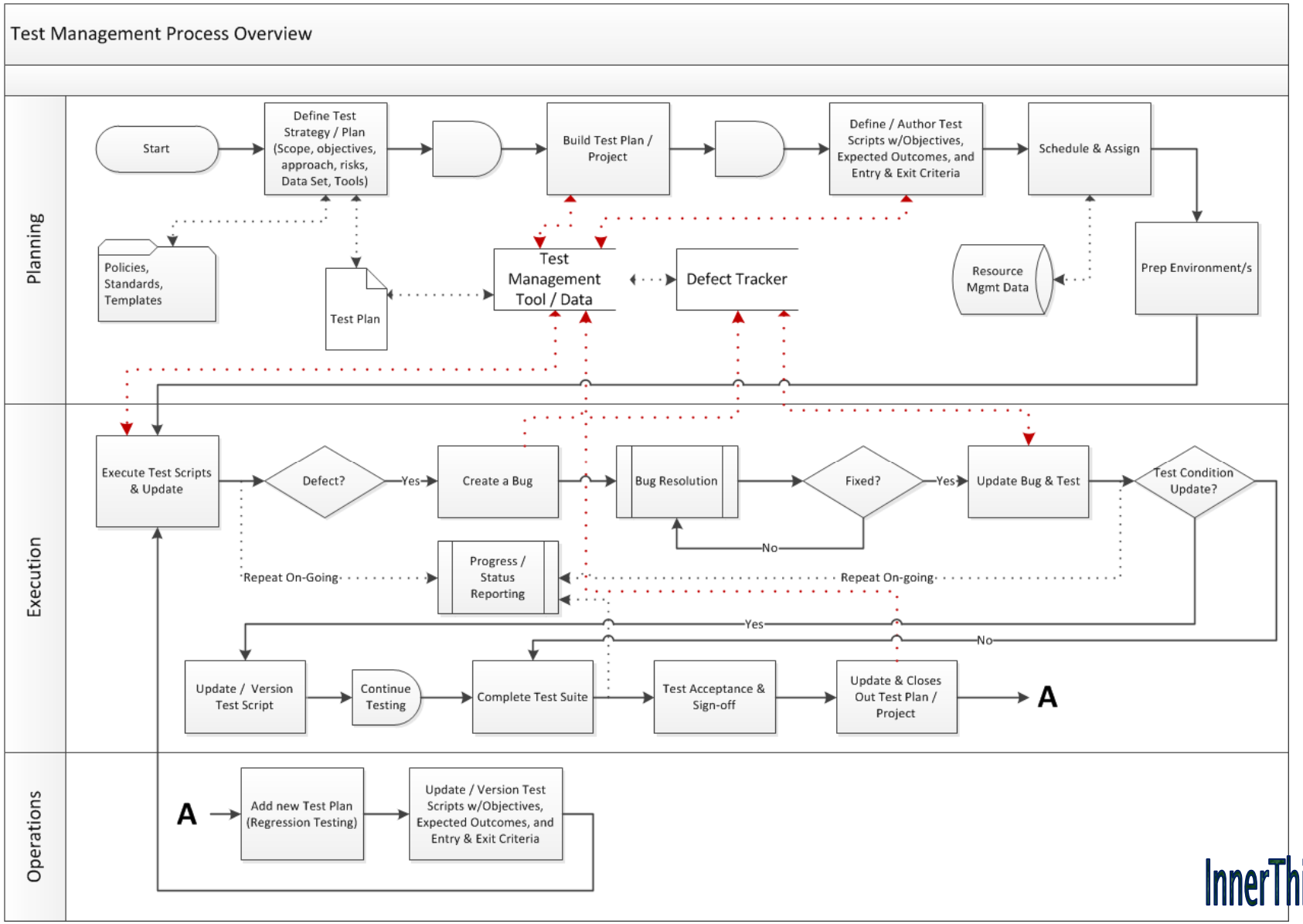
Basic Checklist Going In to Determine level of testing to plan for:

Action / Impact	Code Impact ?	Configuration Impact ?	Functional Process Impact ?	Infrastructure Impact ?	Workflow Impact ?	Policy Impact ?	Integration Impact ?	Security Impact ?
Code Change	UNIT		UAT		UNIT	UAT	INT	UNIT
Configuration Change		UNIT	UAT		UNIT	UAT	INT	UNIT
Functional Process Change	UNIT	UNIT	UAT	SIT	UAT	UAT		
Infrastructure Change	UNIT	UNIT / SIT	UAT	SIT	UNIT / SIT		SIT / INT	UNIT / SRM Review
Workflow Change	UNIT	UNIT	UAT		UNIT/ UAT	UAT	INT	UNIT / SRM Review
Policy Change	UNIT	UNIT	UAT	SIT	UNIT/ UAT	UAT	INT	UNIT / SRM Review
Integration Change	UNIT	UNIT	UAT	SIT	UNIT/ UAT	UAT	SIT / INT	UNIT
Security Change	UNIT	UNIT	UAT	SIT	UAT	UAT	INT	UNIT

Legend:
SIT = IT testing (logic, process, performance, load, security, etc.)
Unit = discrete testing of a specific component, e.g. line of code, or routine, or specific process, etc.
SRM = Security Risk Management
INT = Integration testing (end-to-end)
UAT = User Acceptance testing

InnerThink

Appendix C – Testing Process



InnerThink

Appendix D – Test Framework

Foundational Components:

- Strategic support for Test Management Discipline
- A culture of quality assurance
- Core Policies, e.g. projects must have a define test plan using approved tool sets
- Test Documentation Standards
- Approved, and understood, methodologies
- Defined Metrics
- Defined, and established Test Governance
- Ownership and Sponsorship for practices
- Consistency in test discipline execution
- Procedures over the use of live data, and environments, for testing
- Consistent contractual agreements with suppliers of testing services

Planning Components:

- Resource capacity and capability in place
- Test Strategy definition (objective and approach)
- Definition of scope / coverage
- Configuration management practice in place
- Risks identification
- 3rd Party strategy
- Use of tools & objectives of using
- Entry / Exit Criteria defined

InnerThink

Appendix E – Cont'd

Execution:

- Applied Test Techniques
- Test Phase Containment / Gating
- Entry / Exit criteria evaluation
- Test case specifications defined
- Bug / Defect categorization

Test Reporting & Completion:

- Metrics, progress and productivity monitoring
- Stakeholder Communications
- Defect analysis & feedback into the test process
- Requirements coverage reporting
- Completion reporting
- Test acceptance and sign-off process

- Test Prioritization applied
- Re-usable / repeatable tests
- Inspection, walk-thru's, reviews
- Automated test scripts
- Traceability matrix

Test Admin & On-going Operations:

- Update of regression scenarios
- Regression automation
- Regression test effectiveness
- Emergency fix, tests, and release process
- Live Data use
- Post implementation review process

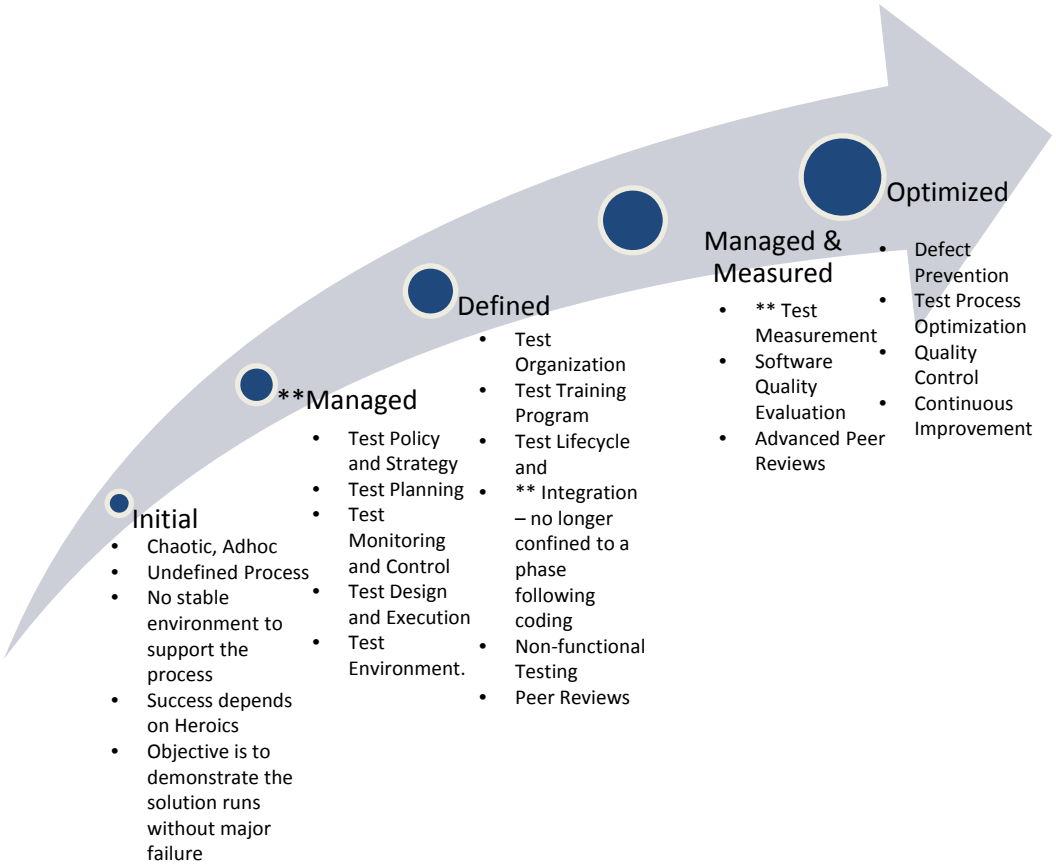
Sources: Deloitte Software Test Assurance Framework Formula

InnerThink

Appendix F – The Maturity Curve

Where are You?

The following illustrates the maturity levels of testing within an organization, where Global IT currently sits, and our desired goals



Source: TMMi, Deloitte

InnerThink