Integrated Testing Workshop – An Innovative Test Planning Approach

Dr. P E Sreenivasa Reddy[1] Sivananda Reddy Elicherla[2] Dr. V Raghunatha Reddy[3] Sivaprasada Reddy Peddareddigari[4]

[1,2,3]Sri Krishnadevaraya University, Anantapur, AP, India – 515003 [4]Bangalore University, Bengaluru, Karnataka, India - 560056

Abstract—Integrated Test Workshop (ITW) primary objective is to help the testing teams, project stakeholders and cross delivery teams to understand the System testing and (or) end-to-end [1] (E2E) testing scope in terms of requirements to be covered, test data requirements, interfaces and test environments that will be used for application or project under test. In a nutshell, ITW is an innovative and test planning approach which will help project test teams to plan their test activities more efficiently than following the conventional test planning approach without much clarity and understanding of the end-to-end functional flow and potential risks involved in the test planning. ITW is not only a planning tool but also static review technique which would help project teams to identify the gaps in requirements and design early in the life cycle.

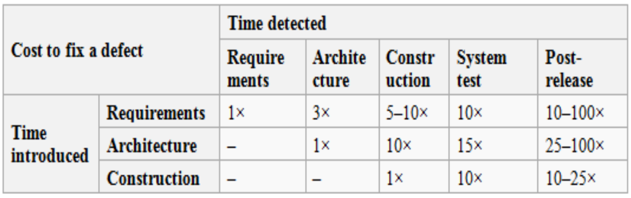
This article will help readers to understand the inputs required to conduct the Integrated Testing Workshop, various stakeholders and their role in Integrated Testing workshop, know-how of conducting the ITW and final outcome of it.

Key words: Test Planning Approach, Integrated Testing

I. Introduction

It has been evidenced that many projects have been failed due to inadequate testing and delivering poor quality software in to the market. The primary reason being, poor test planning in terms of thorough understanding of the scope, lack of application knowledge, control & data flow, data requirements and not arriving at common integrated schedule when there is a overlap between the interfacing applications etc. Also testing team's inability to identify potential risks in test design, planning and execution phases well in advance will put the projects in jeopardy. Thus, it is very essential for an innovative test planning approach like ITW to be in place where all project stakeholders come to the table with a common objective of getting the clarity around what is being delivered or tested and uncover the defects early in the software development life cycle.

Most of us know that cost of fixing a defect later in the life cycle is exponential and It is commonly believed that the earlier a defect is found, the cheaper it is to fix it. The following table [2] shows the cost of fixing the defect depending on the stage it was found. [3]



Cost to fix a defect		Time detected				
		Requirements	Architecture	Construction	System test	Post-release
Time introduced	Requirements	1×	3×	5–10×	10×	10–100×
	Architecture	–	1×	10×	15×	25–100×
	Construction	–	–	1×	10×	10–25×

Table 1: Integrated Testing Workshop

For example, if a problem in the requirements is found only post-release, then it would cost 10–100 times more to fix than if it had already been found by the requirements review. With the advent of modern continuous deployment practices and cloud-based services, the cost of re-deployment and maintenance may lessen over time. A study conducted by National Institute of Standards and Technology (NIST) in 2002 reports that software bugs cost the U.S. economy $59.5 billion annually. More than a third of this cost could be avoided if better software testing was performed [4]. For example if the requirement gaps are uncovered during the ITW will obviously help business analysts and solution architects to do the necessary amendments early in the software development life cycle proactively rather fixing them in later stage of life cycle reactively which would increase the cost of fixing a defect into hundred fold. Please refer the below diagram to understand the cost increase in fixing the defects against the phase that are identified.

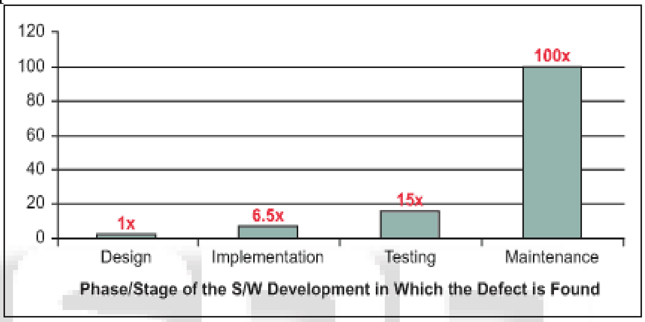


Fig. 1: Integrated Testing Workshop

Hence, innovative approach like ITW will help project teams to identify the defects early in the life cycle.

II. Materials and Methods

A. *Know-How:*

As name implies, Integrated Testing Workshop (ITW) is a lock-down workshop where multiple teams, various project stake holders participate and impart the end-to-end knowledge of an application or project and brain-storm on test planning[5] activities such as scope, test data requirements, potential risks, interfacing applications availability, test schedule and environments etc.

Ideally, the best time to conduct an ITW is when the requirements are baseline and design is complete. It is recommended that, iterative ITW sessions should be conducted for any given project. However, more number of ITW sessions can be conducted when the project is complex and large in nature and required degree of clarity is not achieved in three sessions. Number of sessions required for the project can be determined based on the size of requirements, potential risks, complexity, financial factors and priority of the project. The first ITW session should be conducted as soon as requirements are baseline and the second should be conducted as soon as design is complete. The third ITW session can be conducted post ITW session one and two as to get the required clarifications for the concerns raised in previous sessions.

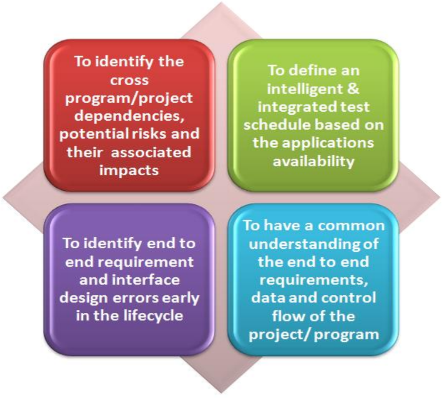
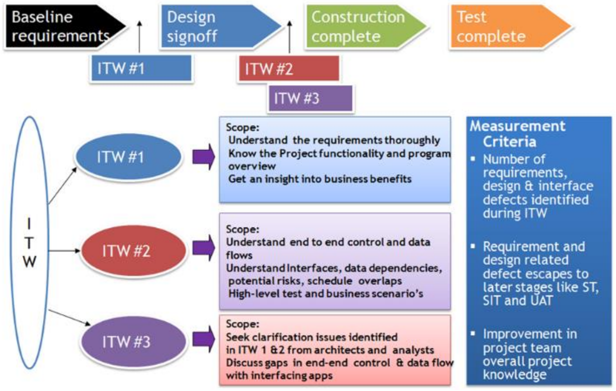

To identify the cross program/project dependencies, potential risks and their associated impacts
To define an intelligent & integrated test schedule based on the applications availability
To identify end to end requirement and interface design errors early in the lifecycle
To have a common understanding of the end to end requirements, data and control flow of the project/ program

Fig. 2: Integrated Testing Workshop

In Integrated Testing Workshop, Solution Architect will present the comprehensive end-to-end data and control flow of the project, business requirements, Interface level transactions and data requirements to the audience which help them to understand and assimilate the requirements from end-to-end and application point of view. Since the cross delivery applications also participating in these sessions test data dependencies between the interfaces and schedule overlaps will be discussed in detail to arrive at the common integrated test schedule. Participants from end-end testing teams will ensure that end-to-end traceability is established with respect to test coverage right from system testing to user acceptance testing (UAT). Also, having conducted a detailed walk through of application level interfaces, transactions and data requirements in ITW sessions will help to uncover the defects early in the life cycle and reduce the defect slippage in the subsequent phases. Program or project test managers will act as facilitators or moderators who will coordinate with required intended audience and note down the action items and ensure that open items are resolved by doing the required follow-ups and by providing appropriate suggestions.

In order to achieve desired outcome from Integrated Testing Workshop sessions, it is important to conduct these sessions in an organized fashion keeping specific objectives in mind. For example, ITW session one should cover the business requirements; program or project scope etc. ITW session two should be intended to cover design documents, control and data flows, interfaces, data dependencies, test cases and so on so forth. The scope for ITW session three will be to uncover the gaps in Interface Documents and any clarifications on requirement issues.

III. ITW PROCESS FLOW, SCOPE & MEASUREMENT CRITERIA



Scope:
Understand the requirements thoroughly
Know the Project functionality and program overview
Get an insight into business benefits

Scope:
Understand end to end control and data flows
Understand Interfaces, data dependencies, potential risks, schedule overlaps
High-level test and business scenario's

Scope:
Seek clarification issues identified in ITW 1 &2 from architects and analysts
Discuss gaps in end-end control & data flow with interfacing apps

Measurement Criteria
- Number of requirements, design & interface defects identified during ITW
- Requirement and design related defect escapes to later stages like ST, SIT and UAT
- Improvement in project team overall project knowledge

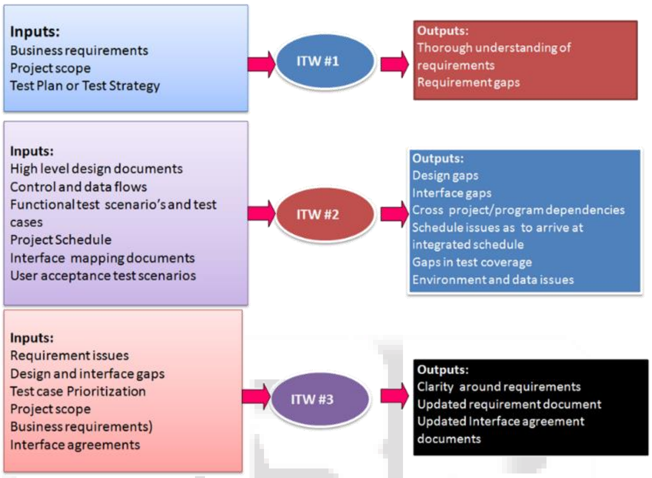
Fig. 3: Integrated Testing Workshop

All these ITW sessions are conducted with specific objectives in mind. Hence, there is set of inputs and outputs are presented or produced for each of these sessions. Also there are set of participants who must be participating in these sessions without fail as their contribution is vital for arriving at the inferences.

Participants for ITW session one and two: Test and Project Managers, Solution Architects, E2E test analysts, system test engineers, system engineering analysts, application developers from all impacted applications.

Participants for ITW session three: Test and Project Managers, Solution Architect, E2E test analysts, system test engineers, system engineering analysts, application developers from the all impacted applications along with Tier1 System engineers

IV. ITW INPUTS AND OUTPUTS



Inputs:
Business requirements
Project scope
Test Plan or Test Strategy

Outputs:
Thorough understanding of requirements
Requirement gaps

Inputs:
High level design documents
Control and data flows
Functional test scenario's and test cases
Project Schedule
Interface mapping documents
User acceptance test scenarios

Outputs:
Design gaps
Interface gaps
Cross project/program dependencies
Schedule issues as to arrive at integrated schedule
Gaps in test coverage
Environment and data issues

Inputs:
Requirement issues
Design and interface gaps
Test case Prioritization
Project scope
Business requirements)
Interface agreements

Outputs:
Clarity around requirements
Updated requirement document
Updated Interface agreement documents

Fig. 4: Integrated Testing Workshop

V. ITW VALUE ADD

Integrated Test Work-shop(s) will not only help testing teams but also help various project stakeholders by providing below value addition.

- To develop common understanding on end to end flow of the projects
- To identify the potential risk and plan for mitigation.
- Understand the data flow and plan for test data to eliminate data dependency during testing phase.
- For early discovery of gaps in application requirement, design and interfaces for end to end flow
- Identify and address the schedule conflicts in code availability and test dependencies
- To define the phased or staggered delivery wherever applicable
- To identify the special needs of the project like volume or performance testing, security testing, etc
- To identify the regression needs for the projects

VI. CHALLENGES IN CONDUCTING ITW SESSIONS

While Conducting the ITW sessions, there will be various challenges that a facilitator would face and need to overcome in order to make ITW sessions more effective. Here are few challenges:

- Teams are virtual and not co-located
- Required input documents are not available or prepared
- Teams do not come prepared for workshop
- Right project stakeholders are not identified or invited to the workshop
- Lack of effective moderation by project Test Managers
- Poor documentation of discussion points and action items etc.

The above challenges can overcome by establishing seamless collaboration between cross delivery teams and educating the teams for preparedness to earn the better results from the ITW sessions.

VII. Results and Discussion

The below data shows that many requirement and design defects have been leaked to the subsequent phases as there was no mechanism like ITW in place to uncover these early in the life cycle (projects A and B).

Static Defects Count before introducing ITW		
	Requirement Defects	Design Defects
Project A	55	36
Project B	45	59

Table 2: Integrated Testing Workshop

Post introducing the ITW in the life cycle (Projects C and D) it is observed that, teams were able to uncover 30% of the static defects early in the life cycle.

Static Defects Count after introducing ITW		
	Requirement Defects	Design Defects
Project A	15	17
Project B	18	16

Table 3: Integrated Testing Workshop

VIII. Summary and Conclusion

Based on the foregoing study results before and after introducing Integrated Test Work shop into the Software development life cycle, it was observed that, ITW sessions helped enormously in identifying the defects early in the life cycle. We have selected couple of projects and analysed the static defect elimination rates before and after introducing the ITW. It is very evident that, by conducting ITW sessions many requirements and design defects have been identified very early in the life cycle.

The below data shows that many requirement and design defects have been leaked to the subsequent phases as there was no mechanism like ITW in place to uncover these early in the life cycle(projects A&B).

Here is the pictorial representation of the analysis pre and post introduction of ITW sessions.

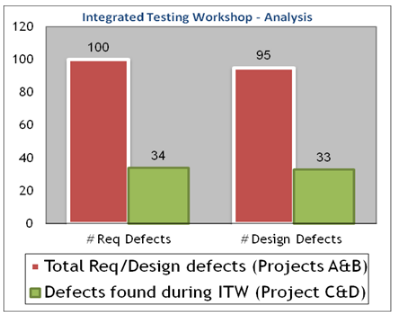


Fig. 5: Integrated Testing Workshop

As you see, the inference can be drawn from the above analysis is that, 33% of the requirement and design defects would have been caught early in the life cycle if ITW sessions should have been employed to the projects A and B.

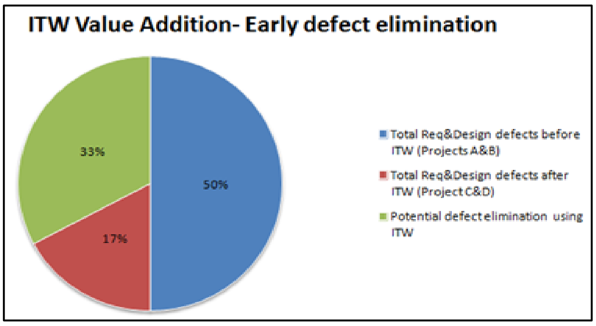


Fig. 6: Integrated Testing Workshop Pic-6

With that said, having ITW sessions introduced in the software development life cycle brought in significant value add in identifying the defects early and also helped testing teams to meet the below objectives.

- To understand the end to end business flow and data flow
- To understand end to end interface gaps and requirement or design gaps and track them to closure
- To identify the potentials risks and find a suitable mitigation strategy
- To arrive at integrated test schedule to tackle the overlaps
- To identify dependencies on the other projects and verify the impacts etc.

Acknowledgment

P E Sreenivasa Reddy and Sivananda Reddy thank the professor T. Sreenivasulu Reddy, Department of chemistry, Muralidhar Rao, Department of Bio-Technology, Sri Krishna Devaraya University, for their extended support. Our sincere thanks due to Mukesh K Kasat, Sr. Executive, IBM, Sarvani Ganesh, Sr. Executive, IBM, Savir K Shetty, Delivery Project Executive, IBM, Shebeen Usman, Delivery Project Executive, IBM, Tushar A Sharma, Sr. Manager, Prakash Kadham, Ravikumar Prabhukumar, Sr. Manager, IBM, Srinivas Vakkalanka, Manager, IBM, Satish Ramamurthi, Sr. Manager, IBM. Last but not the least our sincere thanks to Sreenivasa Reddy’s father Pappireddy Reddenna, mother P. Rani Prameela Devi, wife P. Divya Reddy, daughter Reddy Praharshitha Pappireddy, son Ram Aditya Reddy and brothers P. Vishnuvardhana Reddy, P.Pavankumar Reddy, In-Laws Sri. P.Nageswara Reddy and Smt. Tejavathi and Sivananda Reddy’s wife Parimala, son Dharani and daughter Niayti Reddy for their support and to our colleagues and friends for their continuous support at all the time.

References

[1] IEEE (1990). IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York: IEEE. ISBN 1-55937-079-3.

[2] Wikipedia contributors. "Software testing." Wikipedia, The Free Encyclopaedia. Wikipedia, The Free Encyclopaedia, 12 Oct. 2015. Web. 17 Oct. 2015.

[3] McConnell, Steve (2004). Code Complete (2nd ed.). Microsoft Press. p. 29. ISBN 0735619670.

[4] Software errors cost U.S. economy $59.5 billion annually, NIST report

[5] IEEE Standard for Software and System Test Documentation. 2008. 829-2008 - doi:10.1109/IEEESTD.2008.4578383. ISBN 978-0-7381-5747-4.

[6] “Foundations of Software Testing.” Maximo Quality and Test Community, IBM.com Web. June 24, 2014

[7] “Software testing and quality management” IBM.com Web.

[8] “Test management best practices” IBM.com Web. 07 November 2006

[9] ISO/IEC/IEEE 29119-1-2013 Software and Systems Engineering - Software Testing - Part 1- Concepts and Definitions. ISO. 2013. Retrieved 2014-10-14.

[10] ISO/IEC/IEEE 29119-2-2013 Software and Systems Engineering - Software Testing - Part 2- Test Processes. ISO. 2013. Retrieved 2014-05-21.

[11] ISO/IEC/IEEE DIS 29119-4 Software and Systems Engineering - Software Testing - Part 4- Test Techniques. ISO. 2013. Retrieved 2014-10-14.

[12] Pusuluri, Nageshwar Rao (2006). Software Testing Concepts and Tools. Dreamtech Press. p. 62. ISBN 9788177227123.