

A practical Insight into Framework Testing

Emerging Trends

[CT329]

Author: Ms. Padma Podduturi, Mr. Sudeepta Guchhait

Project: Kuehne + Nagel Project, RT SO (R&D Initiatives)

Email Id: Padma_Podduturi@satyam.com, Sudeepta_Guchhait@satyam.com

Contents

1.	Abstract	3
2.	Terms & References.....	3
3.	Introduction	4
4.	Framework Testing vs. Application Testing	4
5.	Framework Testability	5
6.	Challenges in testing Technical Framework	5
7.	Framework Testing Strategy	6
8.	Implementation of Framework Testing Techniques	9
	8.1 Business Problem	9
	8.2 Approach to Overcome Business Problem.....	9
	8.2.1 Planning.....	9
	8.2.2 Execution	12
9.	Staffing	14
10.	Benefits	15
11.	Best Practices and Tips	15
12.	Summary.....	16
13.	Reference	17

A practical Insight into Framework Testing_CT329

Padma Podduturi, Sudeepta Guchhait

QEDGE, HCU Quality Consulting

1. Abstract

The present day business applications are highly Complex in nature. They need to be of highest quality, and future-proof (Scalable & Reliable).

The application's quality is tightly coupled to the technologies and the frameworks used to build them. Frameworks are like the foundations on which the buildings called applications are erected. As we all know the strength of buildings is a function of foundation, this paper explains the approach for testing technical frameworks.

Framework in this context is a set of technical components providing independent functionality/features. Early detection of performance bottlenecks is also one more key area which is heart of any Testing practice. Therefore, Framework testing; an early life cycle testing, aligned to the development model is the norm of the day and future of Testing practices.

This paper suggests an approach of Framework Testing (FT) by Grey Box Testing Methodologies. FT is expected to test each and every component individually and also test the component when it is been associated with the rest of the framework and business functionalities. Also describes challenges of staffing, best practices and challenges of projects.

Keywords: Framework Testing Life Cycle, Component Testing, Test Approach, White box testing, Code Profiling, Black box testing, Test harness, staffing.

2. Terms & References

Term	Reference
FwT	Framework testing
CIEL	Computer Integrated External Logistic
Fig No:	Figure Description
Fig 01	Framework Testing Techniques
Fig 02	Test Harness Development
Fig 03	Framework Development & Testing Activity
Fig 04	Framework Activity Plan
Fig 05	Single Iteration Activity
Fig 06	Framework Activities in Agile
Fig 07	Framework Activity – Onshore/Offshore Model
Fig 08	Framework Team
Table No.	Table Description
Table #1	Framework testing vs. Application testing
Table #2	Framework Test Methodology
Table #3	Staffing & Skills

3. Introduction

Rapidly evolving technologies are providing Technical Frameworks to develop complex, business applications and thus, Component Engineering is gaining substantial significance in the Software Engineering community. Although there are many published articles addressing the issues in building Technical Frameworks, very few address the problems and challenges in testing components and Frameworks.

Because these Frameworks are complex and use the latest technologies, companies need to employ a vigorous quality assurance program to produce a high-quality and reliable product. Quality teams must get involved early in the product development life cycle, creating a sound test plan, and applying an effective test strategy to insure that these enterprise-critical applications provide accurate results and are defect-free. Accuracy is critical for users who apply these results to crucial decisions about their business applications, their finances, and their enterprise.

Why do we need new approach for testing Frameworks? What are the new issues and challenges in testing technical frameworks? How to find the solutions to solve these problems? What is an effective test process for components? How to measure it? What obstacles we may encounter on the road to test automation of a component testing process?

Test managers and Engineers are curious to know the answers for such kind of testing engagements. But very few articles address them all in one & leave the uncertainties.

This paper illustrates the situations and answers the questions with a practical context to “How an optimized Framework Testing can be adopted”, focused on sub-system level testing activities, were applied to successfully test a critical real-world Java-based Rich Client Application. Section 4: projects the difference between Framework testing & Application testing, Section 5: narrates the challenges in testing technical framework, Section 6: depicts Framework testing techniques, Section 7: Implementation of Framework testing techniques, Section 8: portrays Staffing, Section 9: presents Best Practices & Tips and Section 10: finally summarizes.

4. Framework Testing vs. Application Testing

Many times it is debatable and needs more clarity on how Framework Testing is different from the traditional Application Testing. Below is an attempt to detail more on these two testing practices, which are outcomes and real time observations from some of the projects.

Table#1: Framework Testing vs. Application Testing

Framework Testing	Application Testing
Testing initiated at framework level which is the foundation for building the application.	Testing postponed to the end of application development
Adoption of both white box and black box testing methodologies	Black box testing is aimed and validated for functionality
Since the testing focuses at component level, defects resolution analysis is easier and enables the developer to fix them easily.	As application is built & business rules are applied, it would be costly to trace the origination of defect.
Staffing – It requires the test engineers to be more proactive, technically comfortable.	Test engineers, are required to have only the application functional knowledge
This would best suit for component based software development, Agile mode of Development, hence the future of testing	Testing activities are postponed until the application is completely build.

practices.	
Elimination of Code Performance related defects at the start of test life will cost low	Elimination of Code Performance related defects at the end of test life will cost high
Early detection of performance bottlenecks using the Code Profiling at framework level.	Identification of Performance bottlenecks is postponed to the pre-production stage of the application.
Close interactions with development and testing team is advisable and helps in identifying the scope.	Interactions between the developers and testers are not of high significance.

5. Framework Testability

A Technical Framework is a set of components, which are designed for specific requirement and some are made as re-usable. Also it can be defined as a layered structure indicating what kind of programs can or should be built and how they would interrelate. A framework is generally more comprehensive than a protocol and more perspective than a structure.

Framework Testing implies testing of each individual component for its completeness & correctness. Also it is aimed to test the integration of those components when applied with business rules.

A complete feasibility study should be done to understand the testability of the framework, to be specific, testability of the component.

For example, in typical client server applications components are segregated as client side & server side components. Client side components deal more with the displaying the results to UI and sending the requests from UI to a proxy. Whereas, server side components deals more with business logic, persistence criteria.

Thus, depending on the testability of the components, related testing techniques should be adopted.

6. Challenges in testing Technical Framework

As detailed earlier, Framework Testing itself is an innovative & new in the market. There exist many challenges. Below are few challenges that need to be addressed.

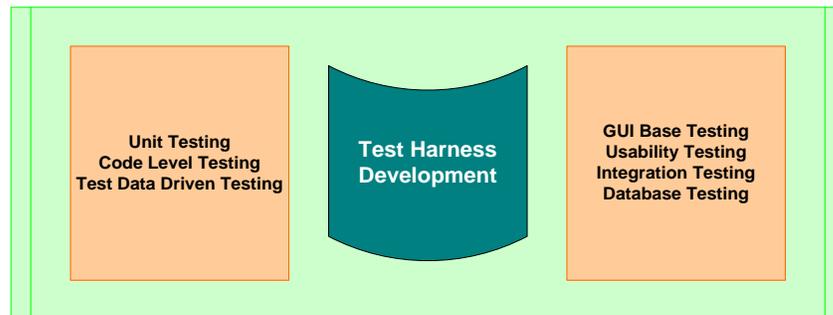
- Resource Skill set: Testing Framework Involves the test engineers to be aware of the technologies been used to develop the framework (Refer – [Staffing](#)).
- Identification of the component and it's testability to consider in scope for testing, component needs to be studied, is it a client side component, server side component, what is the mechanism to test the component (Refer- [Framework Test Strategy](#)).
- Development of stubs, or customized classes to test certain components to understand their re-usability.
- Developer and Tester Interaction especially in Agile mode of Development: Since there is very less time to test the component, it is advised that testing team involves along with the development activities for better understanding the requirements (Refer – [Implementation of FwT Techniques](#)).

- Involvement of multiple vendor and difference in time zone affecting understanding of requirement, releases and test execution, so perfect onshore-offshore model was required (Refer – [Onshore-Offshore Model](#)).

7. Framework Testing Strategy

The cost of fixing a software defect is cheaper if defect is detected in the phase in which it was originated. Efforts spend during testing of Framework is always directly proportional to the efforts spend to address framework issues released during later stages. A new strategy for framework testing is proposed here, which address the gaps, better coverage & assess the quality of Framework code which is later used for developing the Business applications.

Framework Testing = Grey Box Testing = White Box Testing + Black Box Testing
Refer ([Framework Test Development Approach](#))



After certain experiments and researches in one of the projects, a decision about the Framework Testing arrived and is “Grey Box Testing; having in it both White Box and Black box testing’s, ensuring the complete Test coverage of the Framework”.

Table#2: Framework test methodology

Test Objective	Methodology
<ol style="list-style-type: none"> 1. To verify the functionality of the Framework features that were implemented newly 2. Verification of code coverage to ensure the qualitative code to be delivered. 3. Early detection of potential Performance issues at code level. 	<p>Grey box:- White box +Black box methodology is adapted to suit the Test Requirements of framework</p> <p>White Box Testing :</p> <ol style="list-style-type: none"> 1. Features to be tested at code level to validate the functionality. 2. Unit Test execution and Validation to identify the gaps and enhance the code coverage wherever applicable. <p>Black Box Testing:</p> <ol style="list-style-type: none"> 1. Functional validation of Framework features using the Test harness. 2. Code profiling using the identified scenarios on the developed Test harness.

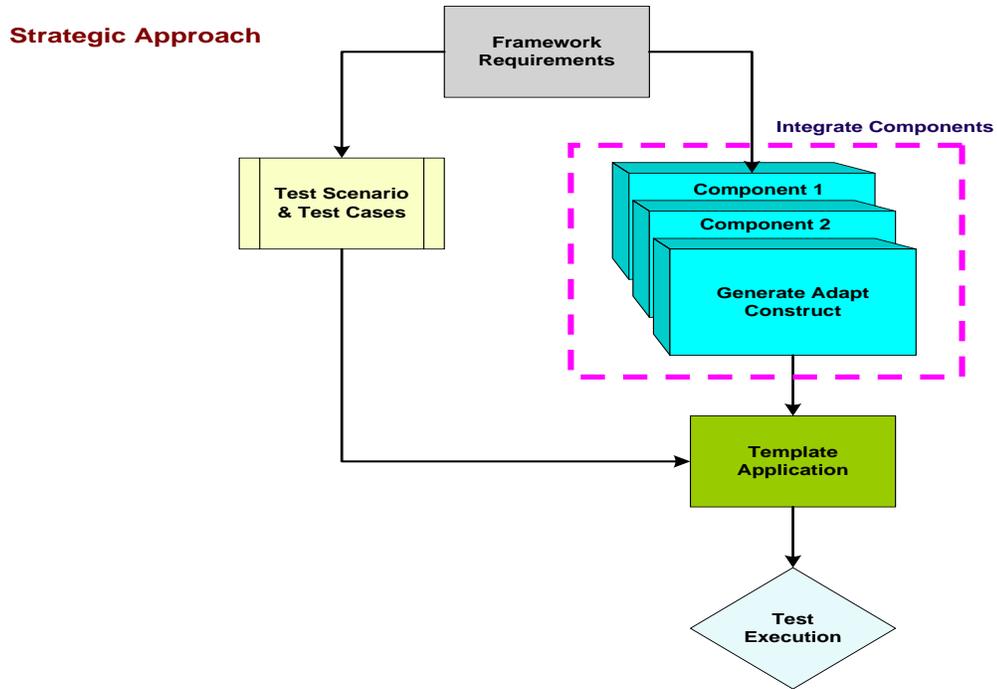


Fig 01: Framework Testing Strategy

Test Harness: A prototype which is developed by integrating individual components by applying business rules. This sample application provides the medium to test the features which requires a UI interface.

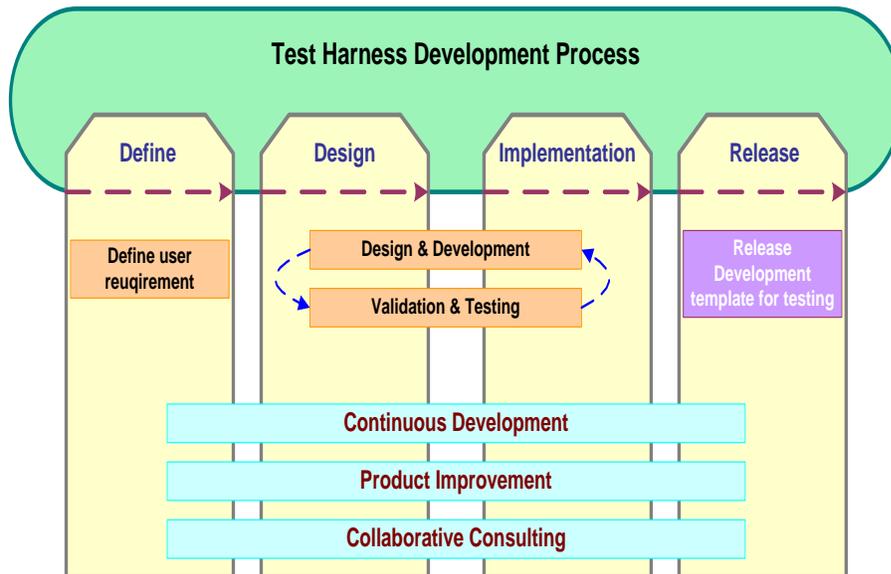


Fig 02: Test Harness Development

Grey Box testing goal can be achieved using Data driven Unit Test validation (White box) +Test Harness (Black box) + Code Profiling (Performance)

White box Testing of Framework: Test objective of White box Testing of Framework is to cover following

- Code level coverage- components to be tested at Code Level to validate their features implementation using the Framework code. Implement the required stubs or custom classes to fully test the functionality of individual components.
- Data Driven Junit Tests – Use developed Unit Test Cases for Testing with identified sets of data to validate and uncover the missing coverage if any.
Suggest the development team to develop the unit test cases for critical functionalities if missed out any.

Black box Testing of Framework: Test objective of Black box Testing of Framework is

- To verify the functionality of Framework features both enhanced and modified using special Test harness.
- **Code profiling:** Compliance to NFR's can be checked using code profiling to identify and measure the code level Performance of framework components at initial stages to assess the Performance bottlenecks and to avoid extra costs at later stage

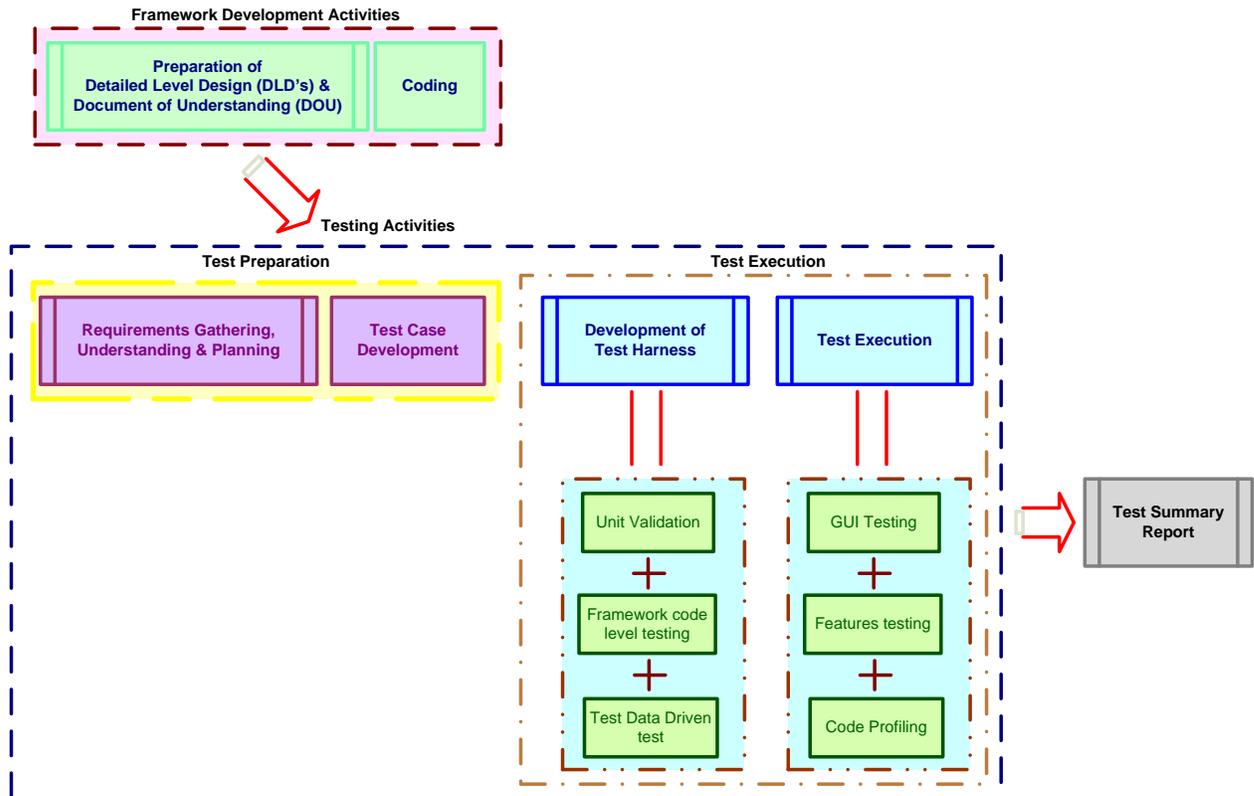


Fig 03: Framework Development & Testing Activities

8. Implementation of Framework Testing Techniques

The above mentioned approach is been currently used in one of the projects. This section details how the FT is carried by applying the Gray Box testing techniques as stated above.

8.1 Business Problem

Customer is the third party logistar company, dealing with Airfreight and Sea freight business. Their business requirement is to move to the “One Global File” concept for all customer and agents across globe. A technical framework which addresses the business problem is asked to be developed using Java Swings to provide the Rich Client and migrating their database from DB2 to Oracle. SDLC adopted is Agile Methodologies where the each sprint (here we call as iteration) is spread for 4 weeks duration and agreed between multivendor involvement.

8.2 Approach to Overcome Business Problem

To overcome the above mentioned business problem and perform complete test coverage test team has implemented usage of “Framework Testing – FwT” during early stages of test life cycle.

8.2.1 Planning

Since there is involvement of multivendor, planning play vital role for development and testing activities. Planning phase has been divided equally for development of framework and testing of framework.

➤ Development planning:

Planning phase for development includes activities like understanding requirement, designing of HLD (high level design) and then proceeding to actual implementation of design converting them to framework code. Once the framework code is dropped in configuration management tool, then code will taken ahead for “Test Harness” planning & development, proceedings to final implementation of components and its features in Test Harness, which will be later used by test team for black box testing.

➤ Test planning:

During the initial phase of development planning, test team will collect requirements for framework development and understand actual requirement for test execution. After complete understanding test strategy will be developed for framework testing and test design activity will be performed. Moment the code drop takes place for framework development, test team continues with test execution flowed by test analysis & reporting. Once the development of test harness development starts test team starts collecting requirement for test harness and prepares a strategic approach to test “Test Harness”. For more information on Test execution part refer ([Test Execution](#)).

Refer below figure 04 for better understanding of each phases of Framework life cycle:
[“Framework Activity Plan”](#)

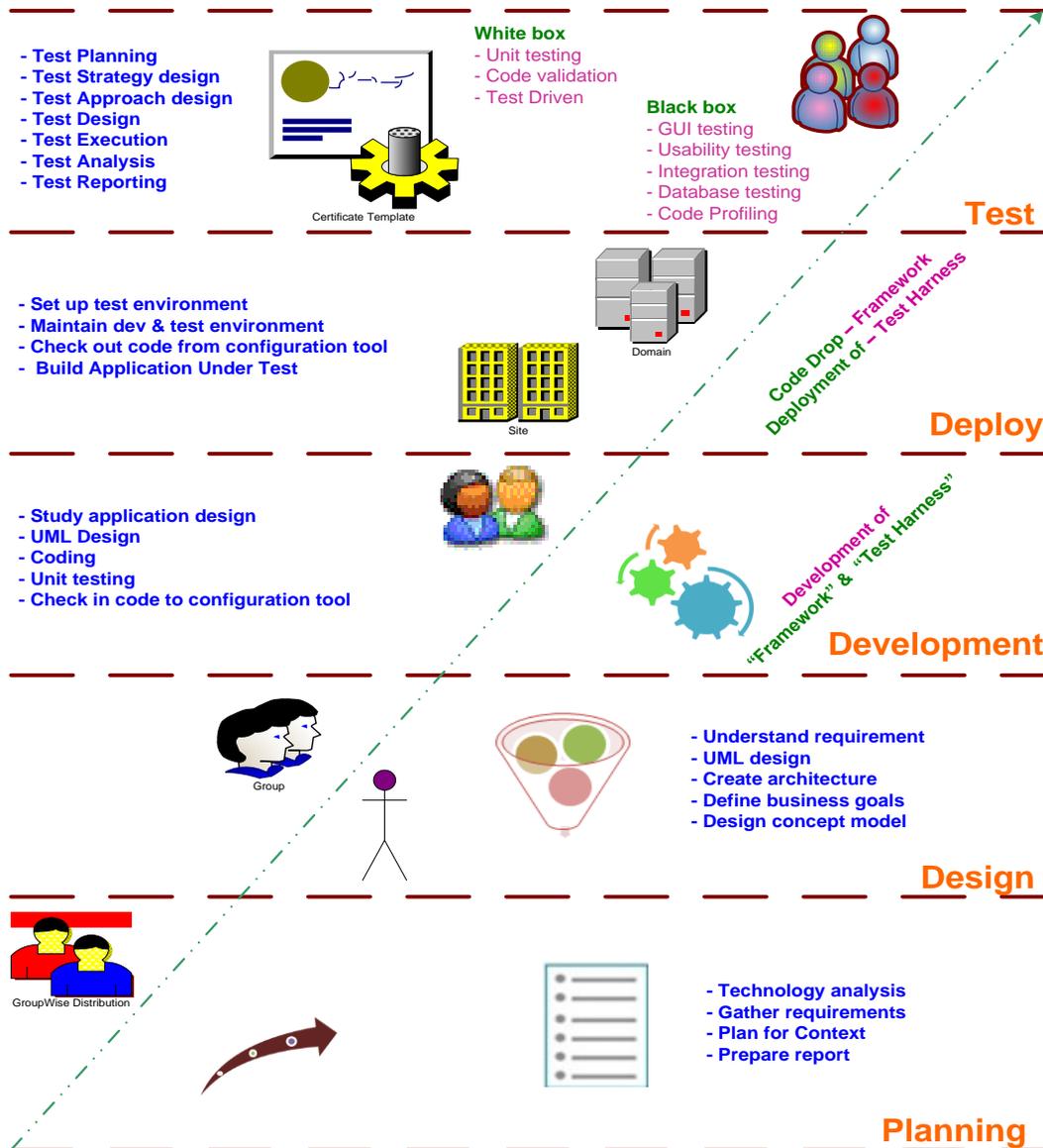


Fig 04: Framework Activities Plan

➤ **Agile model:**

As mentioned above development and testing will be performed by multivendor team, so overcome problems like design, development and deployment and testing in different timeframes by multivendor team a perfect "Agile – Iterative" methodology has been incorporated to be on same page of requirement & understanding. This has helped to development, test and release a product in short cycles removing bottlenecks at early stage of life cycle. One sprint has been considered with 4 releases of build. Each iteration (refer- [Single Iteration](#)) will have parallel activities like Planning, Design and Development of Framework; Test Planning & Execution of Framework ("White Box testing"); Planning, Design and Development of Test Harness; and finally Test Planning & Execution for Test Harness ("Black Box testing"). The complete one iteration would take **4 weeks** of time to complete all activities as show below.

Likewise to complete 1 **Sprint** would take **12 weeks** to complete and release a quarter build featuring all components as planned during Kick-off meeting (refer – [Framework Agile Activities](#))

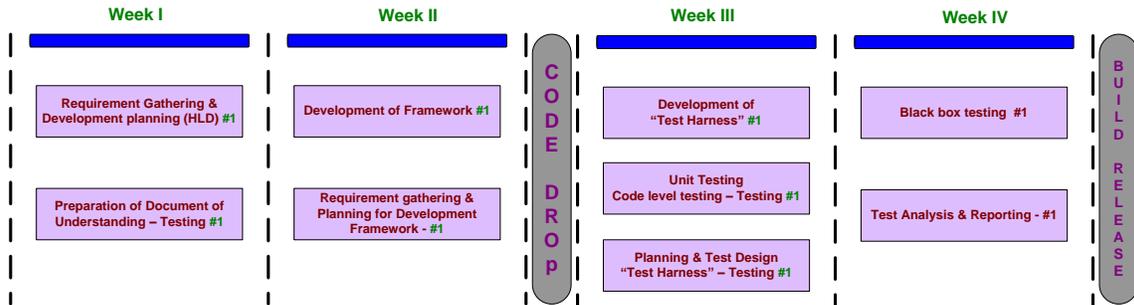


Fig 05: Single Iteration Activity

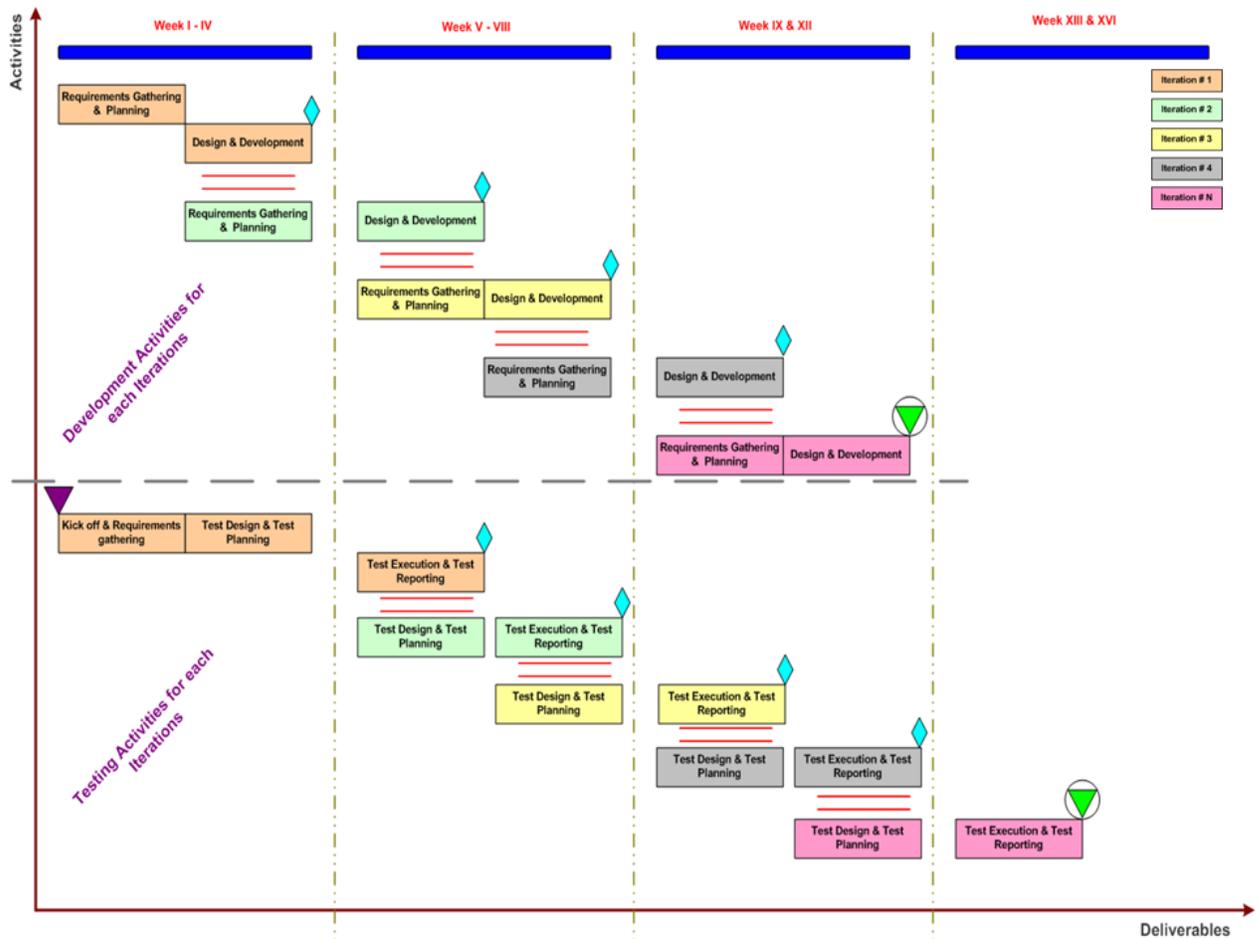


Fig 06: Framework Activities in Agile

8.2.2 Execution

Once the framework development for an iteration is completed, testing team will be ready with the Test Plan, Test Scenario's, Test Cases and Test Data. FwT should be performed using Grey box testing methodology, so need to perform White box testing first and later on performing black box testing on Test Harness.

➤ **White Box Testing:**

By considering the code delivered at the end of an iteration, Framework testing team will adopt the below mentioned white box testing methodologies.

✓ **Data Driven Junit Test Validation:**

Testing team makes use of the Junit test cases developed by the FWK dev team. Those will be tested for various datasets including, null values, boundary values to ensure that necessary exception handling is provided. This process also identifies if any critical part of the code is missed in concluding the code coverage. These observations are then submitted & recommended to enhance the Junit Test cases for the same.

✓ **Code Level testing:**

Custom classes are developed which makes use of the developed components individually and ensures whether the component can be re-usable individually are not.

This process also identifies the unwanted code that has been implemented, objects instantiated but never destroyed which later causes the memory issues.

➤ **Test Harness Development:**

By considering the latest branch code i.e. Code Drop; a test harness will be developed. As per the plan this activity ensures the usage of all individual components into an application facilitating the testing team test the related components and its features.

➤ **Black Box Testing:**

Once the Test harness which facilitates the medium to test all the components is provided below testing activities are conducted.

✓ **Functional Validation of components:**

- During this process components are tested against their behavior when integrated with other components. Data communication between two components when an event is triggered.
- When updates are deletions are performed from the UI the related database is tested for operation validation, ensures that the server side components are responding to the client side components and data is controlled.

✓ **GUI Testing:**

As stated, the application expected to provide rich client using the Java swings. The components are verified for their look and feel criteria's as per the component requirements. RGB patterns, fonts, Borders, Sizes are tested as part of this testing practice.

✓ **Code Profiling:**

Candidate scenarios will be selected to code profile using the DevPartner Java Tool to analyze the following features.

- Performance Analysis – Can be done with respect to expensive operations, methods in which the CPU spends the most thread time and methods that execute slowly because they are waiting

- Memory Analysis -- Analysis is done for most of the methods using the more memory.
- Code Coverage Analysis – Can be done with respect to classes and Methods with the Most of the Lines Not Covered.

➤ **Onshore – Offshore Model:**

Framework development and testing activities has been performed by multivendor and below figure depicts the communication between development team & test team in typical onshore/offshore model.

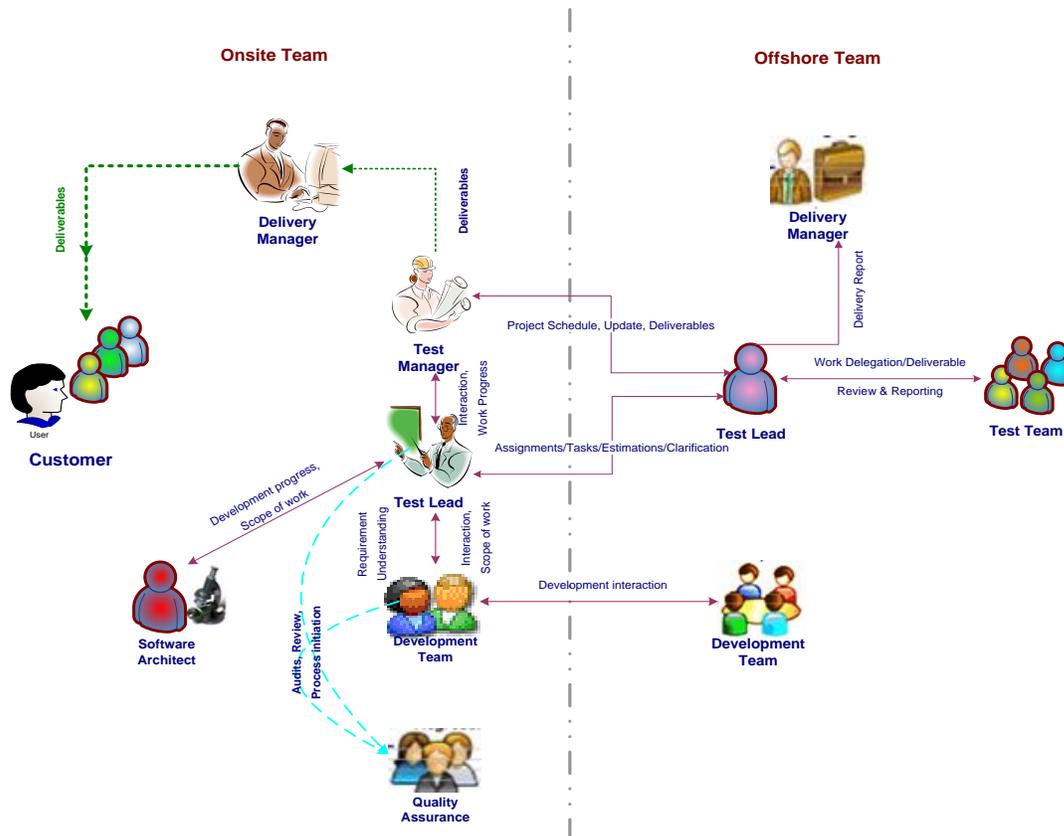


Fig 07: Framework Activity “Onshore- Offshore model”

9. Staffing

As stated above Framework testing activity is not cup of tea for every resource. Framework testing is a niche area of testing, requires well defined skill set to achieve business aspiration.

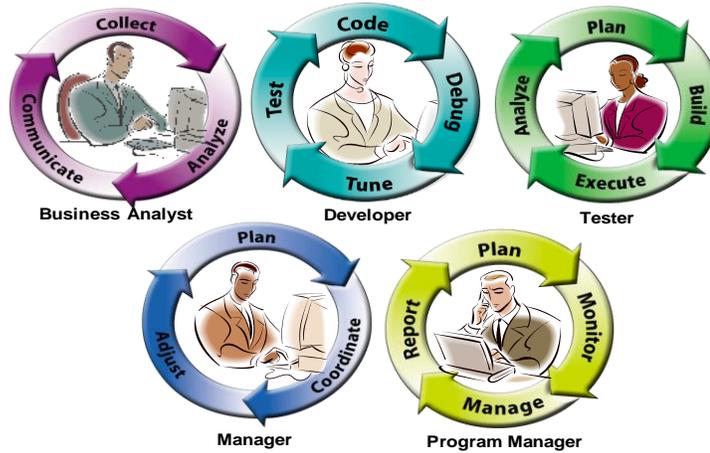


Fig 08: Framework Team

Below mentioned table depicts the skill set required by a resource to work comfortably in framework development & test team. In case if resources don't match these criteria skill set then, project manager need to either recruit skilled resources for respective role or else provide required training to existing resources to continue smooth execution of project.

Table#3: Staffing & Skills

Roles	Responsibility	Skill set
Test Manager	Project management	Testing fundamentals, Planning, Management activities
Test Lead	Project handling, Test Planning, Estimation, Test Design, Test Execution	Testing fundamentals, Test Estimation, Test Management tool, Domain expertise, respective development technology knowledge
Tester #1	Test Design & Test Execution (Unit testing)	Testing fundamentals, Test Management tool, Domain expertise, Respective development technology knowledge, Unit test knowledge
Tester #2	Test Design & Test Execution (Functional testing)	Testing fundamentals, Test Management tool, Domain expertise, Respective development technology knowledge, Database knowledge
Tester #3	Test Design, Test Scripting & Test Execution (Code profiling)	Testing fundamentals, Test Management tool, Domain expertise, Code profiling tools
Developer #1 (Application)	Development of Application and Defect Fix	UML knowledge, Test Management Tool, Configuration tool knowledge, Domain expertise, Respective development technology knowledge, Database knowledge

Developer #2 (Test Harness Developer)	Development of Test Harness and Defect fix	UML knowledge, Test Management Tool, Configuration tool knowledge, Domain expertise, Respective development technology knowledge, Database knowledge
--	--	--

10. Benefits

Following are the benefits by adopting the FT techniques.

- Improved Code coverage ensuring no critical part of code missed from the coverage.
- Early detection of defects thus involves less effort to fix them & developer friendly.
- Early detection of code performance bottlenecks
- Certified reusable components.
- Total test coverage tracing from requirements to features to Test Cases.

11. Best Practices and Tips

Following are the Best Practices and tips that aids to the Framework Testing activities.

- Preparation of DOU's (Document of Understanding) leads to more clarity in understanding testability of the requirements. This avoids the over or under Testing Effort estimations.
- Early Life cycle testing ensures the design level issues to be uncovered & leads to more qualitative framework, thereby, reducing the production costs while an application is built using the framework.
- Publishing the weekly, bi-weekly Test Results to all stakeholders (including dev team) ensures that the defects raised, showstopper defects to be addressed on priority
- Effective Dashboard, Traceability Matrix during each phase (iteration/sprint/duration for Testing activities) ensures the total test coverage.
- Identifying the right resource, right time leads to the successful testing tenure in the project.
- Tollgate Reviews – Since there are many iterative deliverables, it is advisable to have a complete review by , all leads (offshore- onshore), Test manager so that client will be in having a comfort zone & before it is been approved.
- Educate the team & Customer wherever applicable, suggest client if it is required to have better approach. If possible, take to the projects which are adopting the same approach in the organization, where customer gets the assurance before taking a decision.
- Metrics collection – Defects density, Effort Variance, Schedule Variance, Test Execution rate will help and controls the testing activities.

12. Summary

The above proposed Framework Testing approach is optimal and can be adopted in other Framework testing's. It provides approach for total test coverage by testing application components at code level and at functional level using the test harness. Code Profiling is also suggested as part of FwT, which uncovers the performance bottlenecks during the early stages of development, which is the concern area of the organizations to improve the performance of their applications by identifying performance issues in early stages of development.

Testing technical framework is challenging and requires the adequate skilled resources to be part of Testing. The proposed FwT will be best suited in projects implementing agile methodologies.

13. Reference

- [1] J. Langr. Agile Java Crafting Code with Test-Driven Development,
- [2] M. Feathers. Working Effectively with Legacy Code. Prentice Hall, 2005.
- [3] StrutsTestCase for Junit.
<http://strutstestcase.sourceforge.net/>
- [4] Junit Test Decorators.
<http://www.clarkware.com/software/JUnitPerf.html>
- [5] Bytecode libraries.
<http://java-source.net/open-source/bytecode-libraries>.

Authors' Profile

Padma Podduturi (Padma_Podduturi@satyam.com) has been working in Satyam Computer Services for 4.2 years. She has completed B Tech in (Computer Engineering) from JNTU Hyderabad. She is **CSTE** certified and has worked as "Performance Test Engineer" in projects like "**Reuters –UK**", "**MOE-Singapore**", "**AMAT**", "**ALTERA**", "**CITI Group**" and currently working as Test Lead in "**Kuehne+Nagel**" Project.

Sudeepta Guchhait (Sudeepta_Guchhait@satyam.com) has been working in Satyam Computer Services for 4.8 years. He has completed B Engg in (Industrial-Mechanical Engineering) from Nagpur University. He has worked as "Automation Engineer" in "**SAP Labs**" project. Mostly he has worked as "Performance Test Engineer" in projects like "**Reuters-UK**", "**McGraw- Hill**", "**ANZ IT**", "**Kuehne+Nagel**" and presently working in "**R&D Initiative**" of "Regression Testing Service Offerings".