2011 International Conference on Advancements in Information Technology
With workshop of ICBMG 2011
IPCSIT vol.20 (2011) © (2011) IACSIT Press, Singapore

Whitepaper-Actuarial Engines: A business Delivery Challenge?

Amol Parkhe[+], Shilpa Dhumal and Priyanka Godwal

Deloitte Consulting India Private Limited

Abstract. Actuarial Engines: A business delivery challenge is focused on the current industry challenges to test the Actuarial Engine used in Insurance Industries for complex calculations. This paper exhibits industry case studies to show the challenges and then shows how these challenges can be mitigated by phase approaches using the operating model of Actuarial testing (i-ATM).

Keywords: Actuarial Engine, i-ATM, Testing.

1. Introduction

The advance measures and regulations that have been introduced into the Insurance industry have raised a need to use complex and advanced Actuarial Engines. Insurance industry uses different Actuarial Engines available in the market to process Assets and Liabilities to produce Financial Supervisory Authority (FSA), Solvency 2, and International Financial Reporting System (IFRS) compliant results. The Insurance Industry requires a robust and well-defined architecture to support upcoming trends in the industry. The architecture revolves and evolves around these Actuarial Engines.

This paper is written with an intention to address some of the challenges that an Information Technology implementer usually faces while delivering the insurance solution to business. The white paper will address our understanding of why Actuarial Engines are challenge for business delivery? And the approach an Information Technology implementer should follow to minimize its impact on business delivery.

2. Business Case Studies

2.1. Business Case 1

Largest U.S. insurance service provider is using PROPHET as an Actuarial Engine and has implemented a massive data warehouse to support the generation of inputs required by the PROPHET and also to store the outputs generated from the PROPHET. The Information Technology implementer was able to validate everything that is peripheral to Actuarial Engines, but failed to validate the Actuarial Engine itself. When solution was delivered many flaws were identified in the process, design, and in the engine code itself. This lead to huge rework of the solution. Business went through multiple implementation phases to rectify the problem. The results were that the solution was delivered quite late in the stage, which was upsetting for the client.

2.2. Business Case 2

Largest UK insurance service provider is using ALGO as an Actuarial Engine. Business wanted to implement Solvency 2 norms and standards. They believed that everything will hang together and started off with PoC testing. The solution failed at multiple points due to lack of Actuarial Engine testing knowledge skills within the team during requirement analysis and testing phases. Learning's from initial testing enforced the fact that we need to have the required Actuarial Engine testing skills and right composition of the team to

[+] Corresponding author. Tel.: +91-22-61137690.
E-mail address: amparkhe@deloitte.com.

deliver the solution. New test cycles were introduced to address the ALGO testing in detail, including the source data validation.

2.3. Business Case 3

Largest UK insurance service provider is using Excelian as an Actuarial Engine. The firm is using the tool to support Pillar 1 Solvency 2 requirements. The business had an approach to test the Excelian but it was not in-depth. When the system entered into testing cycles, Excelian failed to integrate with other components of the architecture. The integration test approach was taken up to test end to end integration of Excelian along with detailed test approach to test the Excelian results.

2.4. Business Case 4

Largest U.S. insurance service provider is using in-house built Actuarial Engine. The engine is basically a combination of Excel and VBA code. Business plans to decommission the old system and migrate everything to a new engine. The test failed during the migration and business could not decommission the system. The business continued using old solution and kept modifying the new system. Additional cost was spent on maintaining the old system and rebuilding the new system. The business realized that there is a need to spend time in understanding the current source data and the requirements of the new engine. The source data validations and clean-up was not carried out as an initial activity. The new approach was formed to test the source data and the engine in detail.

3. The Analysis of Why It Is A Problem?

Following data points can help us understand why the Actuarial Engines are problem to business delivery:

- Lack of advanced techniques to validate and clean the legacy system source Data
- Lack of well-defined strategies to migrate business from old platform into new platform
- Lack of in-depth test approach to test solutions which are Actuarial Engine centric
- Lack of right team composition to deliver the solution and validate it completely
- Lack of understanding of the engine and its integration with insurers other applications

4. Approach to Address the Problem

As mentioned in the above sections, Information Technology implementer is required to address this challenge in order to successfully deliver the solution to the business. Actuarial Engines play vital role in the system architecture.

The Information Technology implementer usually defines an approach to validate the solution as below:

- Source system validation
- Reporting solution validation

This leaves a gap in the business delivery as Actuarial Engines are not considered in the scope of testing. We would like to propose some approaches to overcome the challenges that Information Technology implementers are facing.

4.1. Approach 1:

Require well-defined specifications for inputs, and outputs of Actuarial Engines and understanding of all interfaces touch points. Following checks can be performed as part of this approach:

- Validating the inputs against specification
- Validating the outputs against specification

- Format checks (data type, data length, table structure)
- Source to target mappings
- Integration between all components of the architecture

This approach does not test the solution in totality and we require further more enhancements to the Approach 1.

4.2. Approach 2:

Require well-defined specifications for Inputs and outputs of Actuarial Engines, understanding of all interfaces touch points, signed off functional and technical specs. The test analyst can perform validations by comparing the old results with new results. Same input data set can be run through existing solution and new solution to validate the results. This approach proves that system is designed as expected and helps us to identify big functional defects well in advance. Certain tools like Excels can be used to compare the results and identify the gaps. The Excel can be designed to mimic the system behavior. Small data set can be pushed through the system and Excel templates, the numbers showing up on the reports can be validated against the numbers on templates. Following checks can be performed as part of this approach:

- Validating the system against specification
- Validating the numbers on the reporting solution

The approach gives a comfort that system works as expected, but is not validating the numbers showing up on the reporting solution. We need few more enhancements to the Approach 1 and 2 to cover 100% testing.

4.3. Approach 3:

Require well-defined specifications for inputs and outputs of Actuarial Engines, understanding of all interfaces touch points, signed off functional and technical specifications, subject matter experts and test analysts who can help with number crunching. After the test analysts performs the comparison the subject matter experts can help quantify the gap between actual results and expected results and also perform report validations and number reasonableness check. The subject matter experts can go through reports and can comment on numbers displayed on the report based on their knowledge and experience.

Following checks are performed as part of this approach:

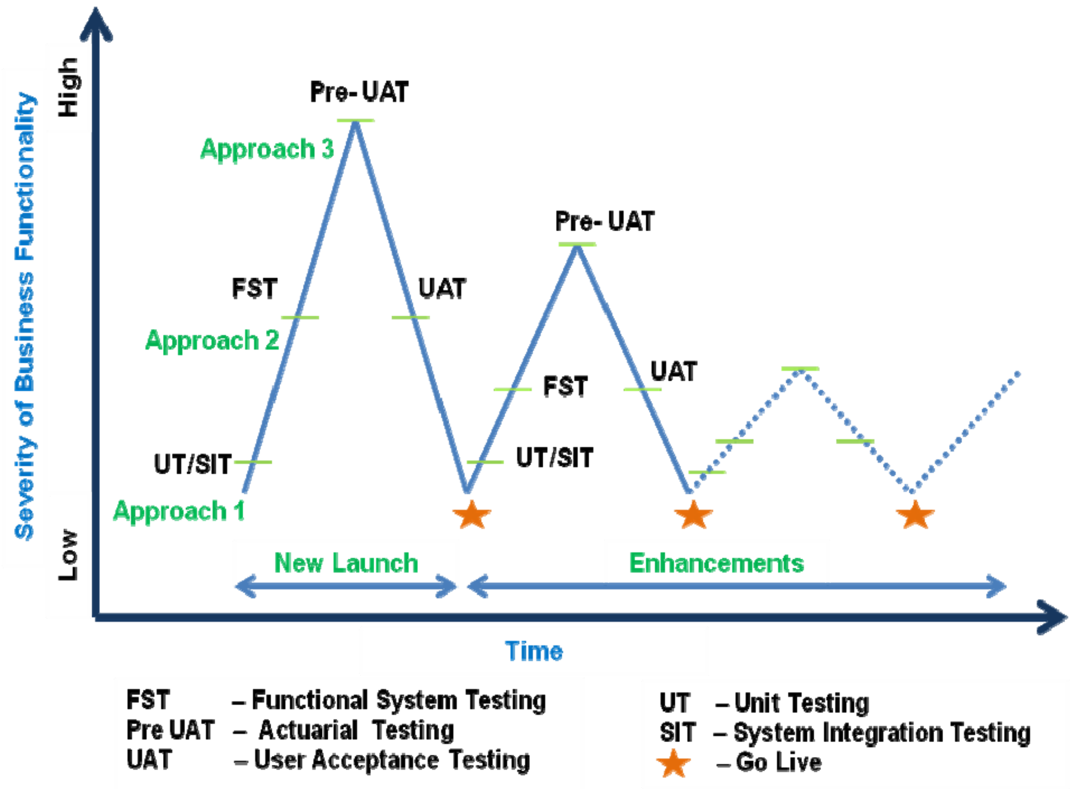
- Number reasonableness check
- Validating the outputs of Actuarial Engines
- Validating the reporting solution correctness

4.4. Analysis of the Above ThreeApproaches

Parameters	Approach 1	Approach 2	Approach 3
Test phases	Unit testing & system integration test	Functional system test	Pre-User Acceptance Testing (Actuarial Testing)
Types of tests	• Input and output validation against specification • Interface touch point validation	• Compare and validate the AS_IS world with TO_BE world results • Test analyst validating the numbers on reporting solution by mimicking the system function in Excel	• Subject matter experts validating the reporting solution • Subject matter experts validating the outputs of Actuarial Engines • Subject matter experts quantifying the gap between AS_IS world and TO_BE world
Team composition	Development team members and testing team members	Testing team members, Business Analyst	Actuaries and Business subject matter experts
Test scope coverage	Approach 1 can support only 20% of testing scope	Approach 2 can support 35% of testing scope	Approach 3 can support 45% of testing scope

Fig. 1: Approach summary.

4.5. Operating Model for Actuarial Engine: i-ATM



Severity of Business Functionality
High
Low
Pre- UAT
Approach 3
FST
UAT
Approach 2
UT/SIT
Approach 1
Pre- UAT
FST
UAT
UT/SIT
New Launch
Enhancements
Time
FST – Functional System Testing
Pre UAT – Actuarial Testing
UAT – User Acceptance Testing
UT – Unit Testing
SIT – System Integration Testing
– Go Live

Fig. 2: insurance - Actuarial Testing Model (i-ATM)

We would like to propose i-ATM - a model for Actuarial Engine testing. The model represents different test approaches and test phases which can be implemented to provide 100% testing coverage for any Actuarial Engine testing. This model compares the severity of Business Functionalities with respect to time for different test phases. X-axis shows the time which represents the sequence of different test phases and Yaxis shows the level of complexity of the business functionalities. This model is divided into multiple sections, first GO-LIVE and subsequent mini GO-LIVE.

The model explains, as the complexity of the business functionality increases, the system enters into different test phases which are resource skill specific. We have aligned the test phases with approaches to highlight that we have an integrated test approach to address challenges. The Information Technology implementer should start with unit testing first and then move into SIT, Pre-User Acceptance Testing phases.

Test Phase	New Launch	Enhancements
	Resource Composition (%)	**Resource Composition (%)**
UT	Development Team – 95% Business Analyst – 5%	Development Team – 90% Business Analyst – 10%
SIT	Development Team – 80% Business Analyst – 20%	Development Team – 80% Business Analyst – 20%
Pre – UAT	Actuaries – 90% Test/Development – 10%	Actuaries – 50% Business - 10% Test/Development – 40%
UAT	Business – 95% Business Analyst – 5%	Business – 95% Business Analyst – 5%

Fig. 3: Test phases with Resource Composition.

5. Summary

Actuarial Engine testing is an integral part of the overall insurance architecture. We have tried to bring forth the importance of Actuarial Engine, the challenges faced during implementations, and finally the operational processes that can be followed to mitigate the risks and challenges that will cross roads during the implementation. We have come up with an explicit test approach and an operating model for Actuarial Engine testing: i-ATM. The i-ATM model is specifically designed to cater the needs and exacting requirements of the Information Technology implementer. The i-ATM process will give the implementer a structured and proven way to drive the Actuarial Engine testing which will fit in the overall business delivery model. To successfully deliver business solutions, it is our recommendation to an implementer to follow and actualize the test approaches and an i-ATM model

5.1. Few take a ways:

- Actuarial Engine testing should be integral part of the overall architecture testing.
- For successful Actuarial Engine testing, the team should compose of the required skill composition of actuaries, business analyst, and test team.
- All the above three approaches along with i-ATM model must be followed to achieve 100% test coverage for Actuarial Engine testing.