

Cost out from financial software product lifecycle: ROI from specialist Training

1.0 Executive Summary

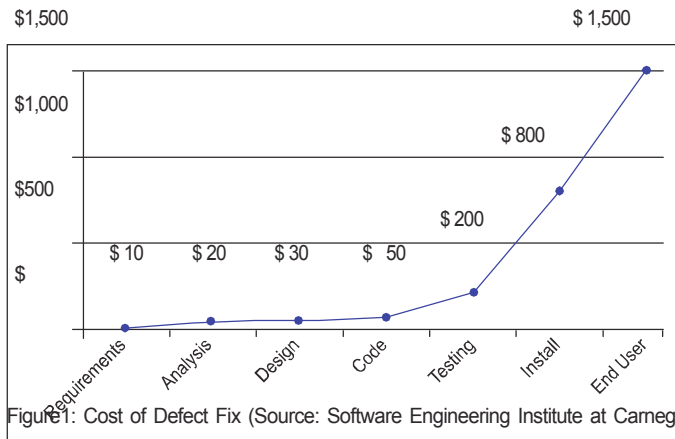
Leveraging Specialist testing, financial and Insurance organisations can potentially look at up to 31% Cost-outs from Software Product Life Cycle Costs.

Component	Costs	Costs
Requirements BR-FS Gap Analysis	3.5%	2%
'First Time Right' Testing	--	11%
Test Automation	19%	5.7%
Right Shoring	40%	12%
Total Cost-Out Potential	62.5%	31%

Table 1: Potential Cost-outs can be up to 31%

2.0 Introduction

It is a well documented fact that cost of defect fix spirals, as one moves through various phases of software development life cycle.



For instance, the cost of defect fix during testing is \$200 while the same during installation would be \$800, a 3 fold increase!. While this may be common knowledge across financial technology professionals, why do we continue to see poor implementations, higher maintenance costs and dissatisfied business users?

For a financial technology leader, faced with the divergent objectives of enhanced delivery quality in tighter budgets, what are the options?

3.0 Costs in Product Life Cycle

It might be relevant to first understand the costs in a typical software product life cycle.

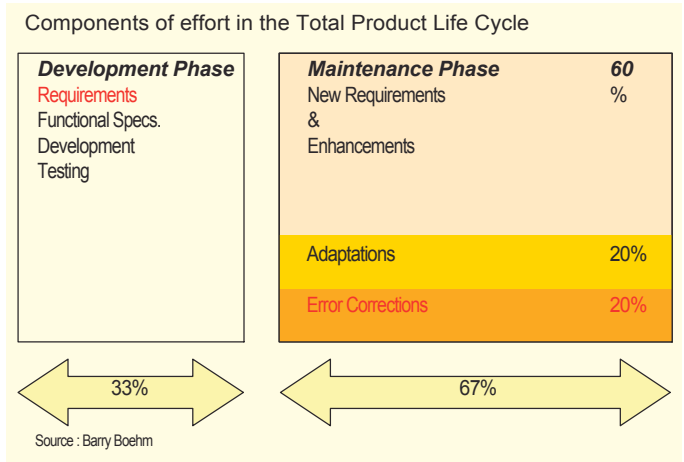


Figure 2: Components of Effort in Total Product Life Cycle

The software product life cycle begins with a business need articulated by the user group. This may be for a new application, a newer generation consolidation of a set of existing applications or for a large-scale redesign and migration of an already operational system to a newer technology environment. The product cycle consists of two parts, the software development phase and the maintenance phase, the relative efforts for which are normally found to be in the ratio 1:2. The initial development phase has distinct specification, development and testing stages to be executed before the application can be deployed. The maintenance phase starts once a software application is deployed. It then enters into useful 'production' and simultaneously enters into a cycle of additional releases driven by defect corrections, functional enhancements and other user-driven changes. Each of these releases has to follow a development, testing and deployment path similar to the original development cycle: a typical application will go through many releases - often 4 to 6 every year - before it is completely replaced by a new software application.

The product cycle is similar for both custom developments as well as for packaged software implementation. In the latter case, the additional development will be driven by customisation as well as interface needs: the whole configuration will then need to be integrated and tested.

As can be seen from Figure 2, around 67% of the product life cycle costs are in maintenance. Of these 20% are due to error corrections. This provides a key insight into the criticality of getting the development, 'First Time Right'.

From an assurance perspective, this means getting the 'Requirements right' as well as getting the 'Implementation Testing right' during the development phase. As the cycle repeats during maintenance mode, getting 'Release testing right' is critical to achieve cost-outs while balancing delivery quality.

4.0 Assurance Partner model for Product Life Cycle Cost-outs

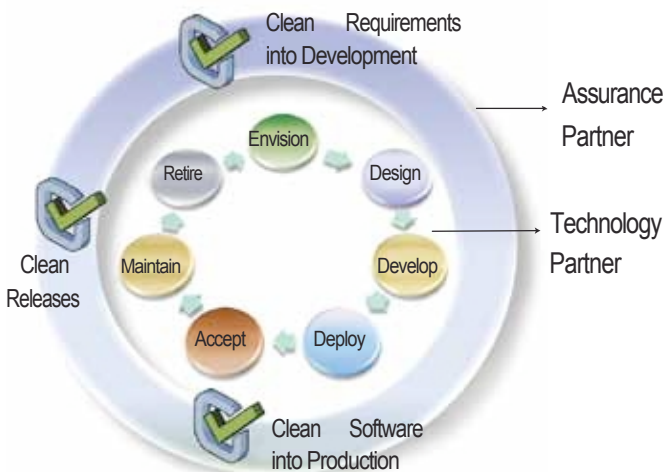


Figure 3: The Assurance Partner Framework

Traditional technology cycle (the inner one) starts with Envision of requirements from the business (right at the top), the solution is designed, developed, deployed and accepted. It is maintained before it is replaced with a new system. The role of assurance is an INDEPENDENT one to this technology cycle. Assurance addresses quality risks on upside and downside.

Upside quality risks come from requirement related issues. Business users give high level, key requirements and expect the technology vendor to fill the gaps, which does not always happen. These gaps could turnout very costly down the cycle. Unstated/incomplete requirements documentation also gives raise to gaps/issues/costly development rework. Developer's understanding of requirements also creates issues as they are ultimately technical people who may not

understand the nuances of financial business processes and linkages.

An assurance partner addresses these upside quality risks by helping clients document the requirements accurately and completely as possible. The partner also reviews the Functional specifications with Requirements to ensure that the requirements are seamlessly translated and every requirement is covered in the FS and covered adequately. This helps save a lot of development rework/testing/ maintenance costs down the product life cycle.

Through this, client can realise 'Clean requirements into development'.

Downside quality risks come from poor testing (which could be either because developer's testing lacks focus and rigor/ business users do not have the skill/time to do thorough testing). This risk can also come from underestimation of testing effort (not using scientific methods to estimate but using gut feel/hunch/ballpark). A specialist testing partner's methodologies, metrics can be brought into test effort estimation and every other activity of testing. The rigor and vigor can ensure that every documented requirement is covered in the test plans. The domain focus brings user thought process into testing.

Through this, clients can realise 'Clean software into Production'.

Once the software enters maintenance mode, areas such as regression are compromised for cost/time reasons. Here assurance partner's expertise in creation of reusable repositories and automated test packs, helps clients balance the budget/time with critical regression testing to realise 'Clean releases'.

In summary, engaging a specialist testing partner for assurance, can potentially impact 62% of the testing costs and 31% of the Product Life Cycle costs. Compelling case indeed for a closer look!

6.0 About Thinksoft

Thinksoft is a specialist in financial software testing with over 8 million person hour track record for global clients in USA, UK, Europe, India and Asia-Pacific. Through our domain focus, structured testing methodologies offshore delivery and test automation expertise, we help clients realise 'business ready software', compress timelines and reduce software product lifecycle cost

www.thinksoftglobal.com
info@thinksoftglobal.co