

STeP-IN SUMMIT 2007

International Conference On
Software Testing

Business Driven Test Automation

By

Sreeja Unnikrishnan and Freethan Meignanam
Infosys Technologies Limited

Sreeja_unnikrishnan@infosys.com and Freethan_Meignanam@infosys.com

Copyright: STeP-IN Forum and Quality Solutions for Information Technology Pvt. Ltd.

Published with permission for restricted use in STeP-IN SUMMIT 2007 in agreement with full copyrights from owner(s) / author(s) of material. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior consent of the owner(s) / author(s). This edition is manufactured in India and is authorized for distribution only during STeP-IN SUMMIT 2007 as per the applicable conditions.

Practices Experience Knowledge Automation

Produced By

STeP-IN
Forum

www.stepinforum.org

Hosted By



www.qsitglobal.com

Abstract

The need for test automation has exponentially increased over the last few years to keep in pace with the frequent changes and enhancements to systems, whatever the line of business. While it is critical that these upgrades to systems be thoroughly tested, it is also crucial to do so quickly to enable faster implementation. Automated functional testing has evolved from the early days of the so called record-and-playback to robust keyword-driven frameworks, resulting in benefits of scalability, better performance and testing, and increased productivity. However these traditional test automation techniques using only functional decomposition and data driven approaches have posed many considerable challenges that prevent achieving the complete benefits of creating and implementing these frameworks.

The response to these challenges was **Business driven test automation**. Business driven test automation enabled the subject matter experts to be an integral part of the test automation process in almost the same way as a technical automation expert. A close watch on the industry trends in test automation shows that the market has rightly moved in to create products and tools that implement the features of *business process* and *reuse*. This has not only reduced dependency on the extremely acute resource pool of automation experts, but also ensured that test coverage is maximized.

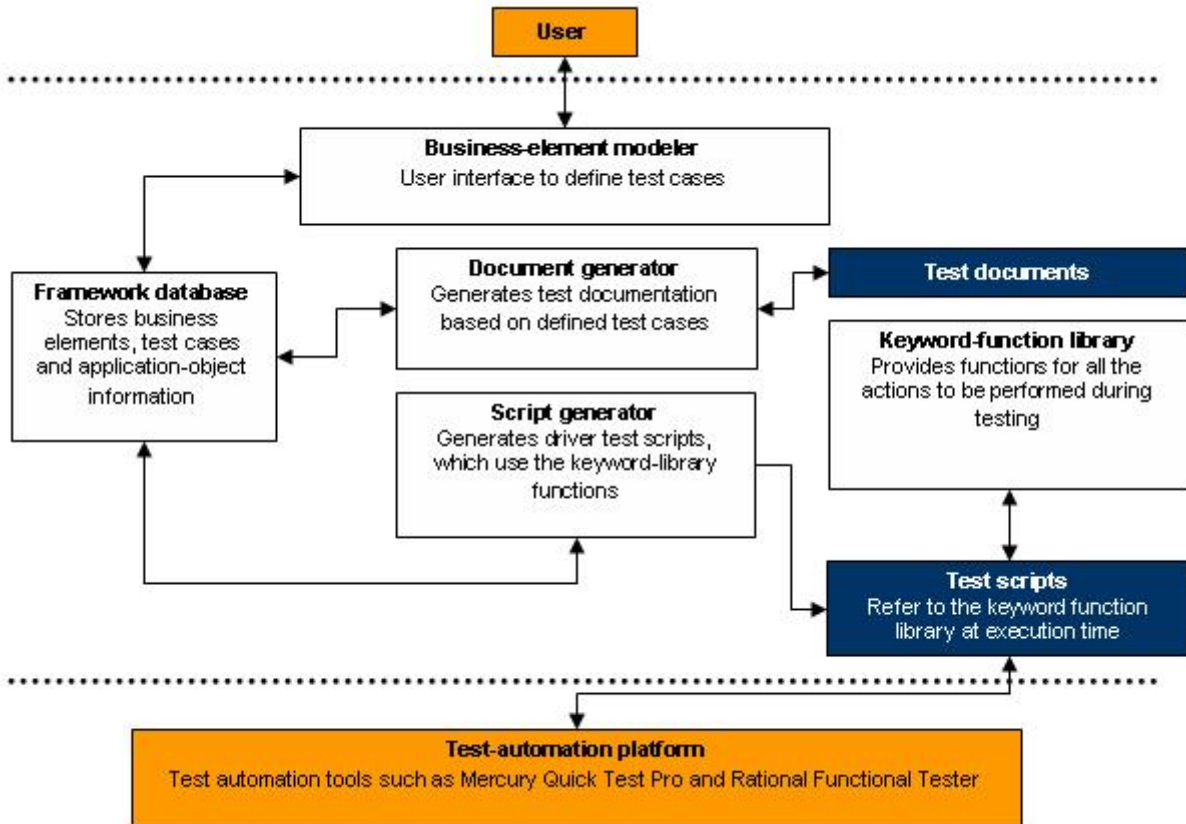
In spite of this encouraging trend, we feel that there still were few gaps in the currently available techniques and methodologies of business driven test automation.

- Dependency on technical expertise
- Limited role of business testers
- Dependency on the availability of the application
- Platform lock-in
- Adherence to documentation standards

The approach that we propose suggests

- **Key concepts** like
 - *platform-independent* test case modeling
 - *business abstraction* layer using Object Logical names
 - *division of labor* and
 - enhanced *reusability*
- **Enablers** like
 - *Business element modelers*
 - *robust scripting engines for multiple platforms*
 - *keyword function libraries* and
 - *Document generators*.

This approach does not attempt to replace the test tools available in the industry, however only augments the test automation process and accelerate it.



The approach has been proved to provide business benefits in terms of

- Very high effort savings (almost 70%)
- Reduction in Time to market
- Lowering TCO
- Increasing ROI
- Improving maintainability

The number of releases to payback the investment made is thus lowered when compared to traditional test automation frameworks.

As the importance of testing and test automation is becoming more widely recognized, faster and more efficient, approaches that speed up implementation of systems and make greater use of the business tester's detailed knowledge of the domain and the application such as business-driven test automation are becoming more popular thereby ensuring that the testing is more effective and thorough.

Importance of Test Automation

During the past decade, companies, irrespective of which domain/field they operate in, have strived to achieve operational efficiencies by adopting various processing systems. The focus has been on automation and moving as close as possible to straight-through processing (without manual intervention).

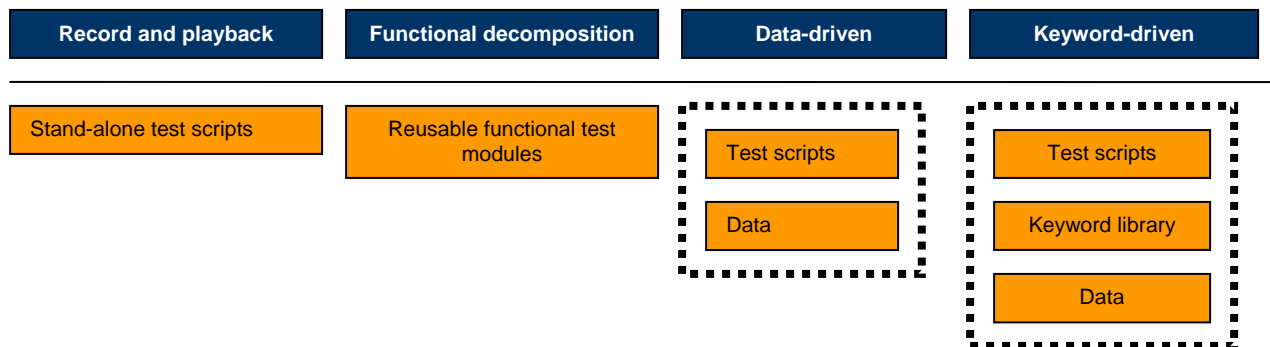
More recently, in the wake of tougher competition and market requirements, companies increasingly have begun adopting so-called next-generation information-technology applications. It has been rightly recognized that it is critical that these upgrades and new operating systems be thoroughly tested – but it is also crucial that they be implemented as soon as possible.

Achieving 100% test coverage after every change, no matter how minor, using manual techniques is virtually impossible. Automation is more effective and efficient, and ensures close to 100% coverage. It also saves time – particularly for the business-acceptance tester who has to perform regression testing.

Evolution of Test Automation

Automated functional testing has evolved from the early days of so-called record-and-playback to robust data- and keyword-driven frameworks (see Figure 1), resulting in benefits of scalability, better performance and testing, and increased productivity. However, each approach has intrinsic problems (see Figure 2), and the most common approach for implementation has been hybrid models.

Figure 1: Evolution of Test Automation



Source: Infosys

Figure 2: Benefits and Shortcomings of Automated Testing Approaches

Approach	How it works	Benefits	Shortcomings
Record and playback	Users' actions are captured, then played back on the application	Ease of scripting; not much technical expertise required	Difficult to maintain test scripts; not extendable; limited re-usability; even small changes to the application require updates of scripts
Functional decomposition	Re-usable, repeatable snippets of functions are created	Modular approach provides some flexibility; maintainable; reduces redundancy; larger test cases can be built in a hierarchical fashion	Data exists within scripts, meaning limited reusability; ease of maintenance depends largely on technical expertise; framework is highly dependent on the application
Data-driven	Input/output data is maintained in external files	Size of the test pack is greatly reduced; improved maintainability	Depends on technical expertise of test team; maintenance and perpetuation are issues
Keyword-driven/table driven	Robust, application-independent, re-usable keyword libraries are built	Ease of maintenance and highly scalable; reduced dependence on application availability	Requires great deal of effort, and is time-consuming; expertise in test-tool scripting language required for framework development

Source: Infosys

Test Automation Challenges

Despite the impressive evolution of test automation, there are still major challenges, including the following:

- **Cost:**
 - Test automation is never a once-off. Test scripts created during the initial automation exercise need to be maintained to keep pace with changes to the relevant application. The cost and effort involved is great for any of the traditional approaches.
- **Dependency on automation expert:**

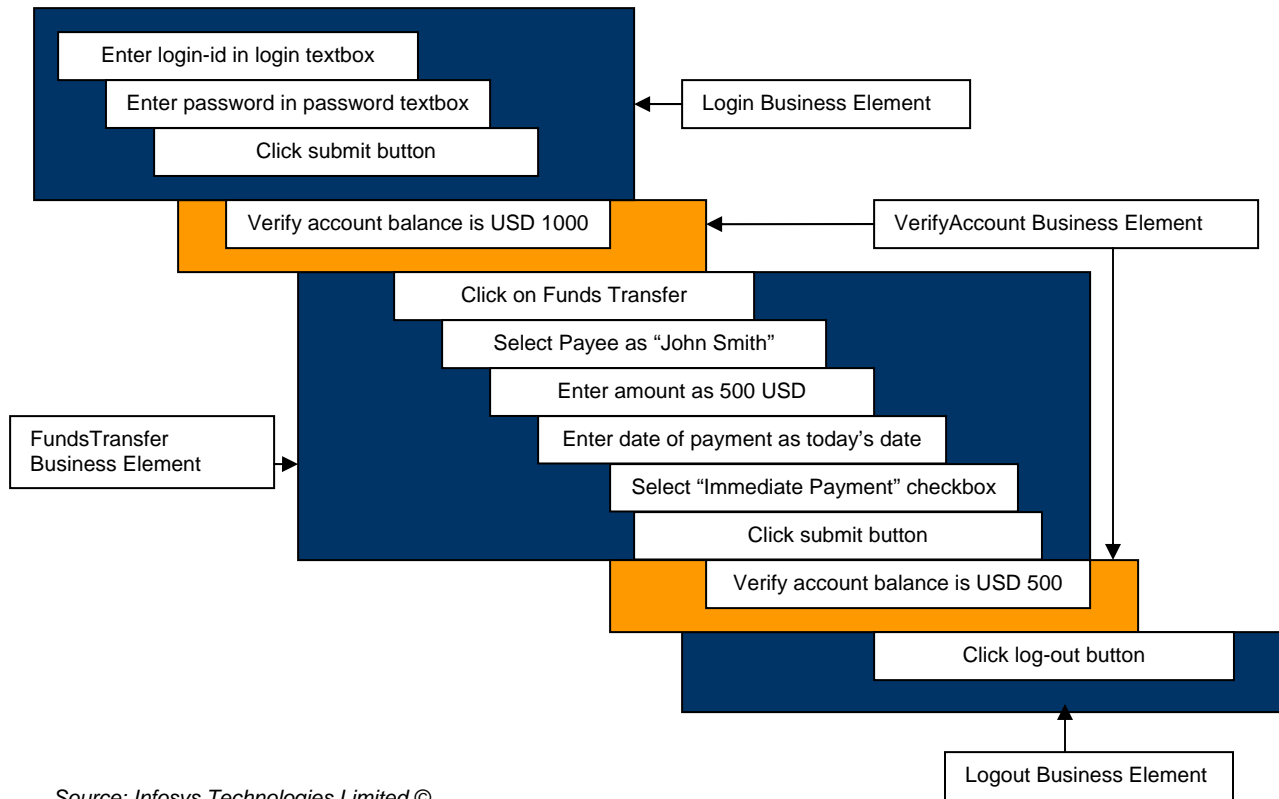
- Test automation is largely a technical exercise performed by a skilled automation expert. Moreover, once the test automation platform changes, we require a completely fresh team skilled in the new test automation scripting language. Moreover, there is no division of labour.
- **Reduced role of business tester:**
 - By contrast, because automation is a technical activity, the users – who typically have a thorough understanding of the application’s functions – have little input, even when scripts need to be used for business user acceptance testing!!! Thorough review of the traceability and completeness of the test scripts is not possible most of the time.
- **Framework design:**
 - Designing and building automation-framework components is a major project, both in terms of effort and time and their effectiveness depends largely on the competency of the creator. Poor design can result in redundancies and very high maintenance costs – and more, re-usability itself might be a question.
- **Dependency on application availability:**
 - Scripting usually starts only after the application is ready (except in advanced keyword-driven frameworks, which are expensive).
- **Platform lock-in:**
 - Traditional test-automation frameworks typically result in so-called platform lock-in once built, which can be a problem if a change of platform is warranted.
- **Documentation:**
 - Requirements must be clearly documented at the start of automation. Any customising or modifications of the framework as per the new requirements, once it is defined is time-consuming.

Our Approach

“The business-driven framework is a flexible test automation framework that enables the business tester to define test cases in English language and generate automated test scripts for a platform of choice at the click of a button based on a pre-defined keyword library.”

It takes the business tester as closer to “script-less test automation“. The framework attempts to empower the business tester with a bigger role in the test automation exercise thereby reducing the extent of dependence on the test automation expert. The composition of the framework allows for it to be effective in any dynamic environment.

The framework is built on a set of key concepts which can be explained using the example below.



The diagram depicts the test steps that constitute a test case for validating an online money transfer transaction. The test steps within this test case are of two kind viz., Action steps and Verification steps.

Business abstraction through logical names:

Test steps are defined using logical names and are aggregated to form reusable business elements (Business Process Classes). The framework provides a user interface where the application components such as screens and controls can be defined using their logical names. Actions such as "Click Submit Button" can be defined on User Interface screens and can then be aggregated to form business elements such as "Login", "FundsTransfer" etc. As the entire test scripting is done using logical names, early start of automation is facilitated even before the application is ready

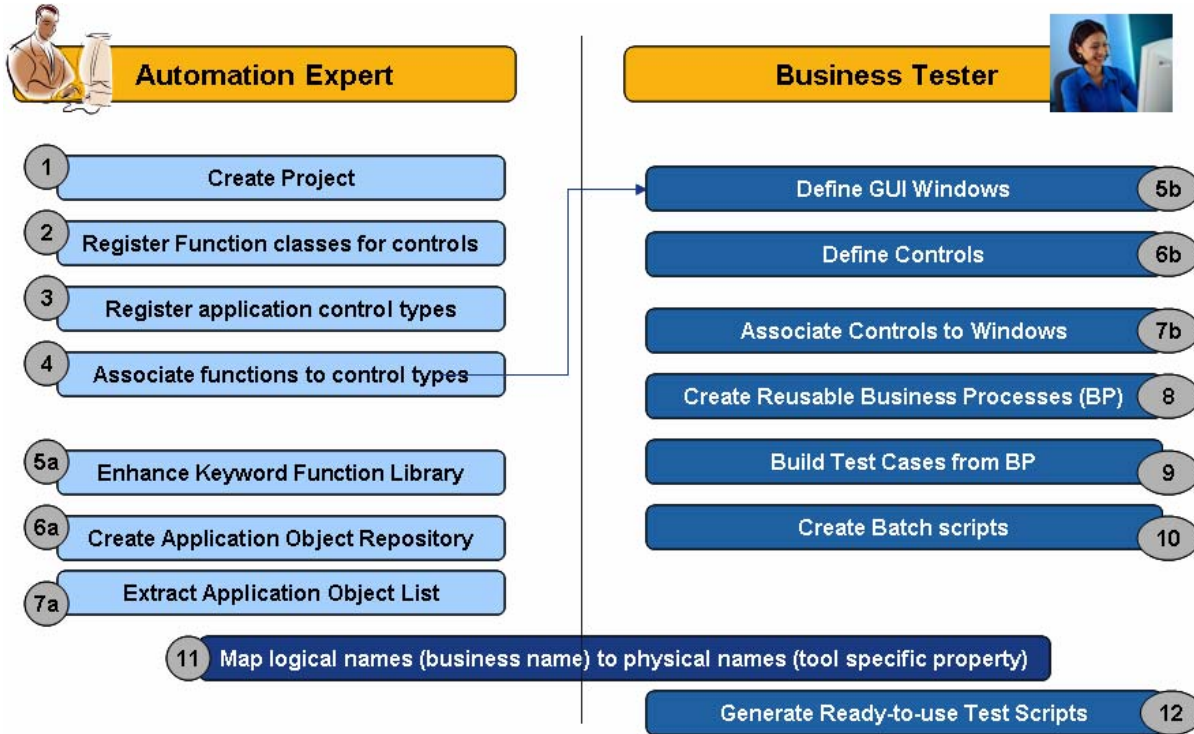
Re-use at all levels:

The approach of decomposing test cases into controls and action, windows and business processes enables reuse at each of these levels. For example, windows and controls can be reused within a business process or across business processes. Similarly business processes can be reused within the same test case (as in the "Verify account" in the above illustration) or across test cases ("Login" will be reused in all test cases). This enables the practice of 'Define once and use anywhere'.

Division of Labor:

The separation of the roles of the business tester and the automation experts enables them to work in parallel resulting in shorter lead-time. The automation expert can define, create and customize the underlying automation library and other technical components while the Business tester creates the Business components. It allows the members from different backgrounds perform activities that they are experts in – encapsulating the complexity of test automation from the business experts who define test automation scripts in simple English language.

The diagram below illustrates the activities for the two roles.



Source: Infosys Technologies Limited ©

Platform Independence:

The availability of inbuilt scripting engines for multiple test automation tools/platforms results in a lot of flexibility. Ready to run test scripts for the same functionality, can be generated for any test automation tool (supported by the test automation framework) with minimal updates (limited to GUI controls/object repository creation and mapping). This enables ease of migration between platforms with minimal rework and organizations can leverage this capability to move to an automation platform that provides the best of breed functions and features.

Enablers

Business Element Modeler:

The web based user-friendly interface allows easy test case definition by the business tester. The modeler has well-defined modules for test case scripting, test script generation and test data definition.

Scripting Engines:

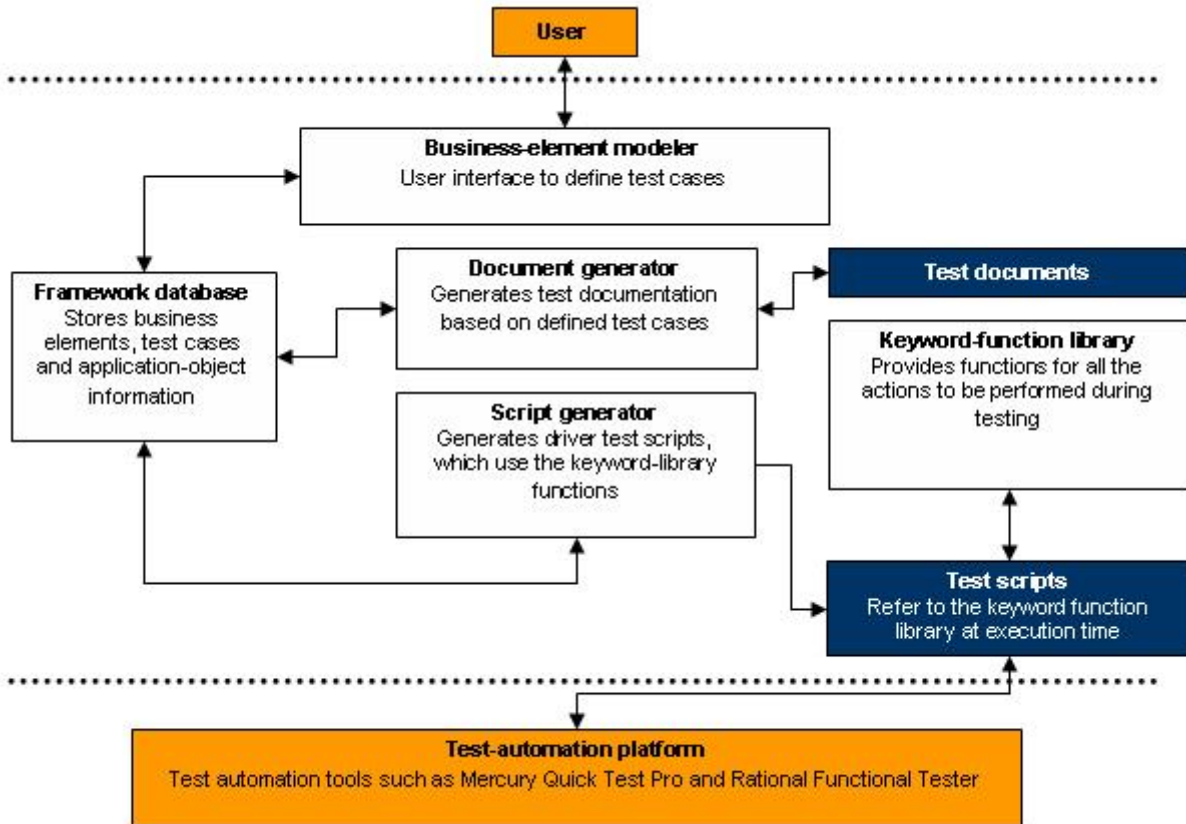
Robust scripting engines intelligently generate 'syntax-ready' and 'ready to run' automated test scripts for multiple technologies and tools like Mercury, Rational, Compuware. The availability of scripting engine makes it possible for non – automation experts to create automation scripts.

Keyword Function Library:

An exhaustive keyword library supports multiple technologies and tools with inbuilt error handling, non UI functions besides all the UI functions making the scripting, even for complex scenarios, much easier.

Document Generator:

Manual test case documentation for every automated test scenario is generated by the automation accelerator solution. Maintenance of these test cases is far easier using this approach as all modifications to the test case are updated to the documentation automatically.



Benefits

Effort savings and Reduction in Time to Market

The availability of an inbuilt robust framework that could be reused for any application and the ease, with which it could be used, reduces the otherwise huge effort and time that need to be expended in the design and development of the test automation framework, creation of automated test scripts and their maintenance. Also the possibility of starting test automation before even the application is ready, adds to the benefits. The average savings in time and effort has been about 70%.

Lowering TCO

The relative ease of customizing the test automation framework for a particular application and the huge reduction of the test suite maintenance effort reduces the total cost of ownership thereby enhancing the economic value for the investment made.

Increasing ROI

The number of test cycles required to break even the effort and other investments would be greatly reduced using this approach, thereby resulting in very quick pay back. For even very complex applications, the Return of Investment has been achieved in ~5 or lesser cycles.

Improving maintainability

The business driven test automation framework enables incorporation of the changes to the automation scripts with very minimal effort - from a user interface- that any non-automation expert can handle. Additionally, the simple modular structure and adherence to standards and best practices improve the maintainability of the automated scripts to a very great extent.

Accelerated process

The business-driven framework simplifies and accelerates the test automation process. The acceleration is made possible by removing the dependencies allowing the business/functional tester and the automation scripeter to work in parallel.

Conclusion

As the importance of testing and test automation is increasingly being realized across all domains, newer approaches, such as the business-driven test automation, to accelerate the process would find greater acceptance. Efforts are being taken by all stakeholders, mostly the test automation tool vendors to anticipate the needs of the business for test automation through a business-driven framework and constantly improvise them. As a result of this shift in the approach, test automation would outgrow the confines of the technology teams and expand into the business users' domain and the focus would be as much on the business tester as on the testing technologists to ensure that the application changes are tested and implemented faster than ever before in the dynamic world.

Author Biography:

Sreeja Unnikrishnan is a Test Analyst with Infosys Technologies Ltd. She has 4 years of experience in evaluating test automation tools and defining and implementing test automation frameworks. She currently leads the development and deployment of packaged test automation solutions from Independent Validation Solutions (IVS), the testing group of Infosys



Freethan Meignanam is a Test Engineer with Infosys Technologies Ltd. He has been involved in developing and testing ready-to-market solutions involving test automation frameworks for the last one year.