

Executive Bloggers	Visual Studio	Application Lifecycle Management	Languages	.NET	Platform Development	
--------------------	---------------	----------------------------------	-----------	------	----------------------	--

Microsoft Application Lifecycle Management

Learn to use Visual Studio, Visual Studio Team Services, Application Insights and Team Foundation Server to decrease rework, increase transparency into your application and increase the rate at which you can ship high quality software throughout the application lifecycle

Case Study: Using froglogic for test UI automation with Java based applications and Microsoft Test Manager



November 17, 2013 by [Charles Sterling](#) // 1 Comments

0
 0
 0

In just the last month i have been asked **three** times for a case study or examples of how to test Java based systems using Microsoft Test Manager.

Knowing one of the ALM MVPs, [Marcel de Vries](#), is finishing up a very large Java project i reached out to him to find out how he just accomplished this.

Despite the fact he was super busy with the Visual Studio launch in the Netherlands and [Visual Studio Live](#) in Florida he still managed to pen a blog post for the ALM Blog.

Thank you Marcel!!!

In my daily work as an ALM consultant I get interesting challenges once in a while to help out teams implement ALM with team Foundation Server. This was also the case when I was asked to help SNS real insurance with their implementation of TFS across all their software development teams.



SNS Reaal Insurance had selected Team Foundation Server as their primary ALM toolset. For them the most important they reason selected



[Download Visual Studio](#)

[Download TFS 2015](#)

[Visual Studio Team Services](#)



Search MSDN with Bing

Search this blog Search all blogs

Share This Post

Tags

[Administration](#) [Agile](#) [ALM](#)

[Announcement](#)

[Automated Testing](#) [Azure cloud load testing](#) [Coded UI Test](#) [Coded UI Testing](#) [Code Visualization](#) [Customizing](#)

[Debugging](#) [DevOps](#)

[Diagnostics](#) [Extensibility](#)

[Feedback](#) [Git](#) [Guidance](#) [insights](#)

[IntelliTrace](#) [Load Testing](#) [Managing](#)

[Technical Debt](#) [Manual Testing](#) [Performance](#)

TFS is that it provides a central repository for all disciplines involved in the software lifecycle including the feedback loop back from production back to the development team. They had already decided to move to a more Agile and Lean approach of developing software, but their expanding list of single purpose development tools without integration or holistic view of the over all development lifecycle, was becoming a bigger problem each day.

The following statement described what we got asked to do:

“Please help us to increase productivity and predictability of the software development cycle. Enable the road to a Lean IT organization by implementing one tool that supports the whole software development life cycle”

Team foundation Server was selected as the tool to support all teams. The first team we were asked to onboard on Team foundation server was a team that used a set of common off the shelf products used for the basis of the SNS REAAL pension fund. This is a common strategy these days, with more and more companies moving away from building all their software custom for their business. With this strategy most of the work involves integrating in their IT landscape with all the required external interfaces. The team was about 60 people in size and they asked us to help them in three areas:

1. Implement kanban for their teams.
2. Implement software configuration management
3. Most importantly to them, implement a test automation strategy so they could start deliver more frequently.

The posed an interesting question: Since the off the shelf products they selected was a Java Based solution how best to integrate them?

For Version Control, automated builds and Work Item tracking – no problem!

-But for test automation there are some challenges to overcome. One of them being the fact the out of the box solution for UI test automation from Microsoft is CodedUI and this technology does not support Java Based User interfaces for test automaton.

So the big question we got to solve was, how can we build a robust test automation layer, using tools that integrate with TFS and still enable us to test Java based User interfaces?

froglogic Squish for java based UI automation



After consulting with the Microsoft Product team, we got in touch with a German based company called Frog Logic. Frog Logic provides a whole suite of test tools, including tools to test java User Interfaces. While there are many other tools in the market place that can just test Java Based UI's this tool was special as they offered build integration of their test tools with TFS out of the box!

If you look at what Frog Logic has to offer, then one of the most important things that helped us, was the ability to write the tests in the IDE the Java developers were already using, called Eclipse. Froglogic integrates with both Eclipse, and the TFS tools. Meaning that all scripts we wrote to automate the tests, can be placed under version control and the software check-in would automatically integrate with work items. This would keep our end-to-end traceability in place AND solve our problem of testing Java Based applications.

In Figure 1, you can see that the tools in the eclipse IDE contains Java Script based automation scripts that we can check in and out of TFS. The fact we choose Java Script, is because that is more common these days to use and many developers know how to write in Java Script. We could also have chosen for other languages, since frog logic support a whole range of non-proprietary scripting languages like Perl, Ruby,

[Process Templates Rangers](#)

[Release Management](#)

[SonarQube Team Foundation](#)

[Build Team Foundation SDK Team](#)

[Foundation Server](#)

[Testing Test Management TFS](#)

[TFSservice Unit Testing Version](#)

[Control Visual Studio Visual](#)

[Studio 2010 Visual Studio 2012](#)

[Visual studio 2013 Visual](#)

[Studio ALM Visual](#)

[Studio Online Visual Studio Team](#)

[Services Work Items](#)

Archives

[July 2016 \(1\)](#)

[June 2016 \(20\)](#)

[May 2016 \(17\)](#)

[April 2016 \(18\)](#)

[March 2016 \(24\)](#)

[February 2016 \(13\)](#)

[January 2016 \(22\)](#)

[All of 2016 \(115\)](#)

[All of 2015 \(186\)](#)

[All of 2014 \(173\)](#)

[All of 2013 \(154\)](#)

[All of 2012 \(207\)](#)

[All of 2011 \(81\)](#)

[All of 2010 \(28\)](#)

[All of 2009 \(1\)](#)

[All of 2008 \(5\)](#)

[All of 2007 \(30\)](#)

[All of 2006 \(76\)](#)

[All of 2005 \(22\)](#)

Subscribe Blog via Email

Subscribe to this blog and receive notifications of new posts by email.

<input type="text" value="Email Address"/>	
<input type="button" value="Subscribe!"/>	<input type="button" value="Unsubscribe"/>



[Back to top](#)

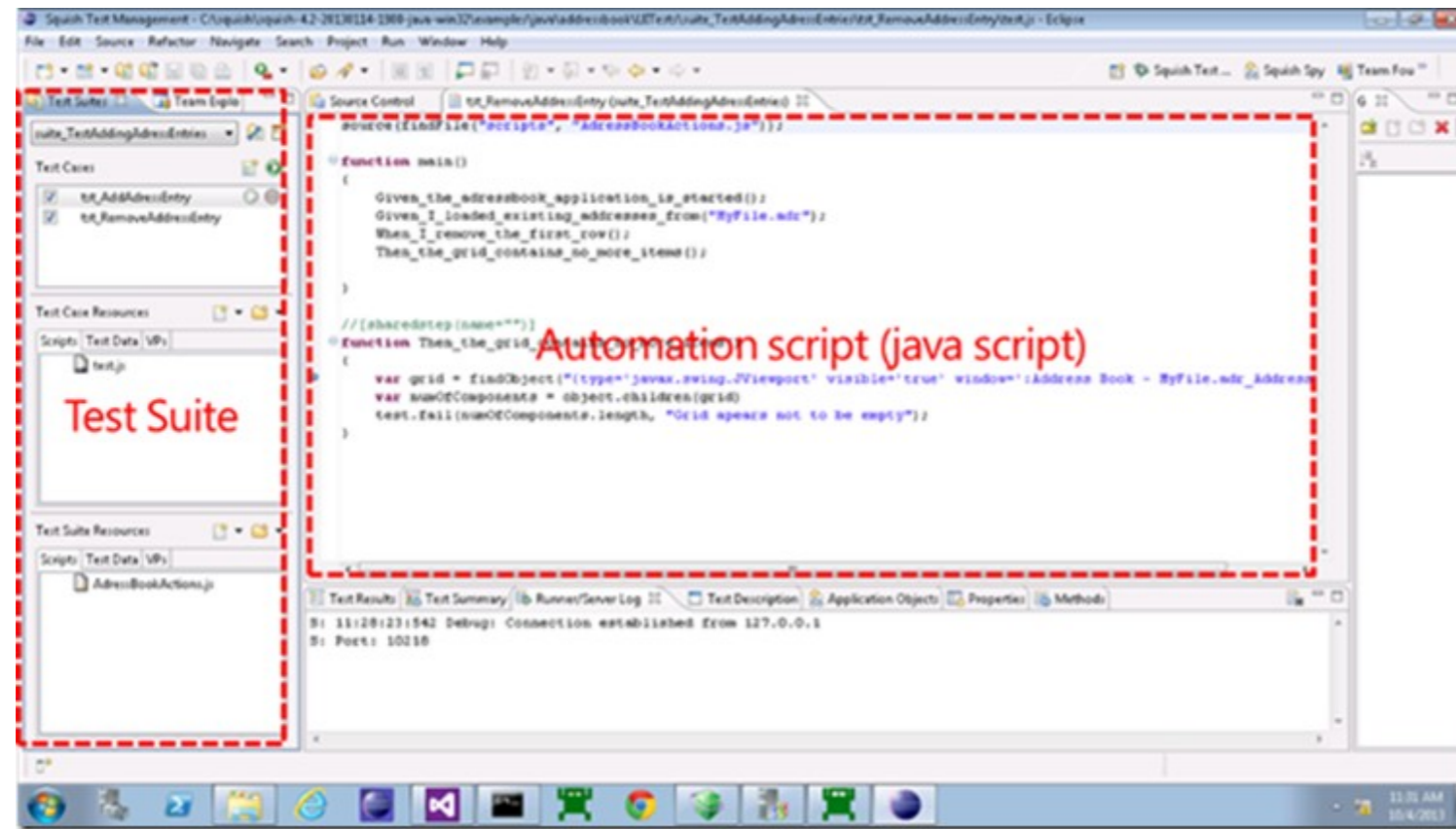


Figure 1: Eclipse IDE for Test suite development

Once you have developed your test suites in Eclipse, you can create the test cases you want to run in Microsoft Test Manager (MTM). These test cases will help you track the tests against your requirements, or stories that are delivered by the development teams. The MTM tooling will keep track of the test cases, enable you to schedule test runs and gather test evidence on how tests ran and what the results were of those test runs. MTM enables e.g. video capture during your test runs, so when something fails, you can review later what went wrong and it solves part of the non repro issue many developer and testers encounter each day.

Once you defined the test cases you can open them in Visual Studio where Frog Logic has a plugin. This plugin then enables you to register the test automation script with the test case, the same way as you would have done for a codedUI test. You can see in Figure 2, that you have the test suite loaded in Visual Studio, and that we can run the tests straight from Visual Studio as well. It even integrates with the agile runner infrastructure introduced in Visual Studio 2012. This is also the place where you associate the tests with the test cases you defined in MTM.

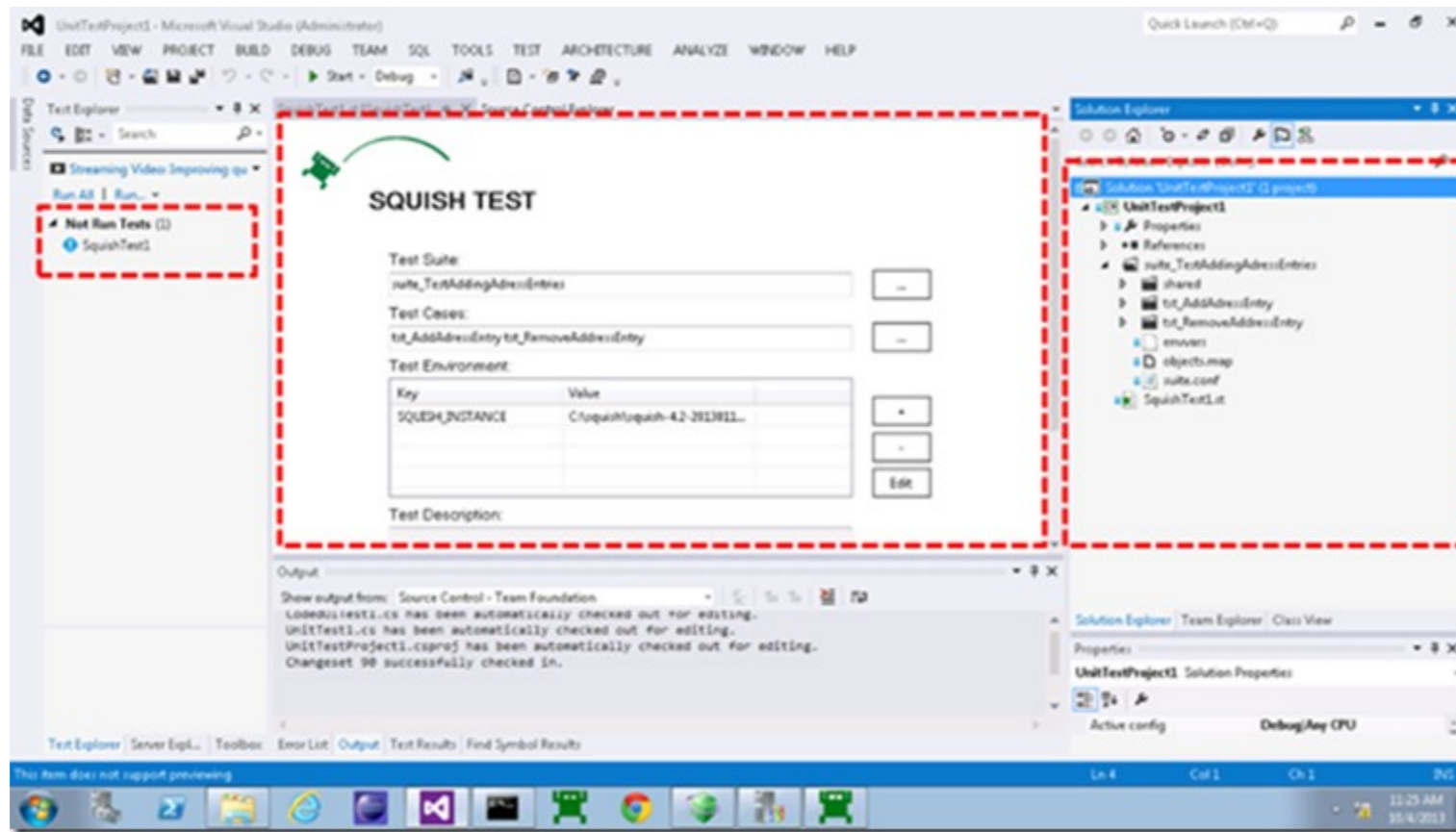


Figure 2: Visual Studio integration to link to MTM

In figure 3, you can see that the test automation is registered with the test case, and this now enables us to run the automated test using MTM and the standard Lab infrastructure. This means we just registered a physical environment in lab management, and this enabled us to have the functional testers run the automated suites as part of their daily test runs.

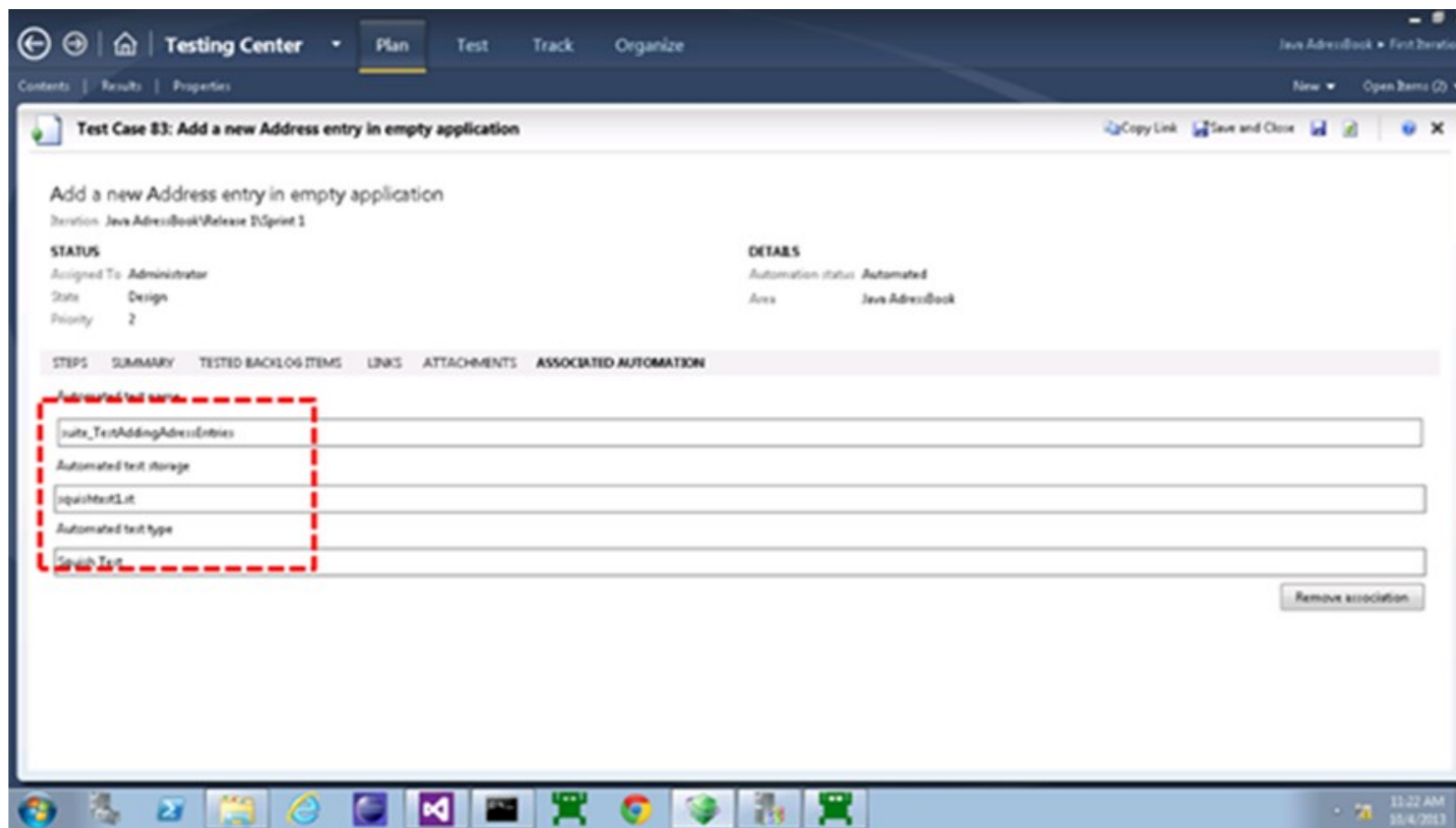


Figure 3: Run your test automation from MTM

You can take it one additional step further and that is integrate it in the team build environment. This means you define a lab management build, where you select the environment, the test plan and the build you want to test (e.g. latest build of the product) and then the build can be scheduled to run each night or even at each check in if you want.

As you can see the froglogic tooling can help you solve a lot of the problem of automating tests that need to test different technologies including Java. They have more solutions in place, but the one we started to use at SNS REAAL was the Java Desktop testing tools Squish by froglogic.

Conclusion

The most important learning for our team was that the test automation need could have been solved many different ways, but we needed a strategy that offered a holistic view of the development process, integration with TFS and solved our main goals. We could have picked many of the other tools that enable Java application testing, but the fact froglogic enabled seamless integration and reporting to out of the box, was for us very important. Now we can leverage TFS for the one and only point of truth of our development process rather than dozens of the other single purpose and siloed tools.

I hope this write-up gives you an impression on how we solved a common problems haunting many large development teams and how froglogic provided us with a solution that fully integrates with Team Foundation Server giving the developers SNS REAAL a fully integrated end to end solution.



About the Author



Marcel de Vries is a Technology Manager Info Support for the Netherlands with more than 15 years of experience in professional software development using the Microsoft development platform. Building enterprise information systems for fortune 500 companies most of them situated in the Netherlands. Expert in Application Life Cycle Management and improving software development practices in the enterprise. Microsoft awarded ALM MVP and Regional Director for the Netherlands.

Microsoft Regional Director and Visual Studio ALM MVP



Tags

[Test Management](#)

[Test Recording](#)

[Testing](#)



Back to top



Roger Keigner

3 years ago

Chuck, how does this compare with Test Architect? Just curious...i thought test architect was the primarily partner for test automation with Microsoft Test Manager.

Comments are closed.



Back to top