

ALM Mag.com

Empowering Rapid Delivery of Quality Software

abstract-aqua-wave--backgrounds-wallpapers

TFS Migrations

2008 & 2010 to 2012

Audio Podcasts

Technology & Process with the Experts

ALM Consulting Update

Industry Case Studies

Industry Surveys

**Interviews with
Industry Leaders**

Best NEW Features

**Event
Wrap Up
& Review**

**More
BONUS
Video**

TFS -
Beyond
Source
Control**

Index Page

Newsflash !!!

A promotional banner for Visual Studio 2013. The left side has a purple background with the Visual Studio logo and the text "It's here" in large white letters. Below this, it says "Visual Studio 2013 Preview" and "Download now" in a black button. The right side features a black square with colorful fireworks and a red circle with a white right-pointing arrow. Below the arrow is the number "12" in red.

Visual Studio

It's here

Visual Studio 2013 Preview

Download now

→

12

Features

- 04** ***Welcome to ALM Mag***
Editor's Welcome 

- 05** ***Lean Software Development in a Nutshell***
Examines the realities of software development and concisely explains the Lean approach to addressing those realities
by Al Shalloway 

- 14** ***Lab Manager Demo***
A speedy demo of Lab Manager with SCVMM and the basics of the Lab Manager Build Template
by Steven Borg 

- 15** ***ALM State of the Industry Reports: Which One is right?*** 
A look at the latest Gartner and Forrester reports focusing on ALM toolsets
by Clementino de Mendonca
- 19** ***EclipseCon Review*** 
A review of several of the presentations at the March EclipseCon event
By Keith Pleas
- 25** ***Microsoft ALM Certification – Part 2 of 3*** 
Part 2 on Microsoft's new ALM certification focusing on
Exam 70-497: Software Testing with Visual Studio 2012
by Anthony Borton
- 29** ***Release Management with TFS 2012*** 
There are many features in TFS 2012 that support creating a release management strategy
by Martin Hinshelwood
- 35** ***TFS Admin Reports delivered to your inbox daily*** 
TFS Admin Reports emailed daily notify and alert on cube status
by Anna Russo
- 37** ***What's New in TFS 2012? – Team Explorer*** 
New to TFS 2012... this article focuses on new features in Team Explorer
by Mohamed Radwan
- 44** ***Countering 6 typical arguments against the Agile/Lean approach*** 
The arguments against and mitigating rationale for an Agile/Lean approach
by Tomáš Tureček

Welcome



Keith Denham
Publisher

Treasure Cove Publishing
Volume 1 – Issue 4
May/June 2013

ADVISORY BOARD:

Sam Guckenheimer
Group Product Planner for
Visual Studio & Author
- Microsoft

Keith Pleas
Organizer & Conference
Chair - ALM Summit

EDITORIAL REVIEW:

Anthony Borton
ALM Ranger
ALM Trainer & Consultant
- Enhance ALM Pty, Ltd.

Mike Vizdos
Agile Coach
Certified Scrum Trainer

Clementino de Mendonca
National ALM Architect
- Neudesic

Subscriptions:

almmag.com/subscribe

Email Address Changes:

support@almmag.com

Mission Statement:

"Our mission is to promote,
support and empower
rapid delivery of quality
software."

This issue coincides with the 5th named version of Team Foundation Server, TFS 2013. This version brings exciting new features and functionality to play that many have been waiting for, particularly those wanting an easy tie from the executive planning level into the development tier.

Al Shalloway examines.

presents another great Northwest Cadence video demo of.

Clementino de Mendonca examines y.

Anthony Borton's article is the second of a three part series outlining the new Microsoft MCSD ALM certification. Part 2 covers:

Exam 70-497: Software Testing with Visual Studio 2012.

This is Microsoft's first exam focusing primarily on Microsoft Test Manager (MTM) 2012.

Martin Hinshelwood tells that to the contrary, TFS 2012 has many features that support creating a release management strategy.

Mohamed Radwan presents the new and interesting TFS features available in Team Explorer.

Not yet a member? Take advantage of our early bird Offer... click this link for more details. <http://www.member.almmag.com/goto/foundingmember>

Enjoy!... and Warmest Regards,
Keith Denham

Keith Denham

P.S. Don't forget, this is an interactive magazine, tap on the audios and videos and tap on the highlighted links to see additional related content.
Tap on the Index Page Button at the foot of any page to return to the Index.

Scott Ambler



Martin Hinshelwood
ALM Consultant
[Northwest Cadence](#)

W hile

A professional Development Team is supposed to create potentially shippable

The Principles of Agile Architecture



**Alex Yakyma and
Dean Leffingwell**
[Scaled Agile Framework](#)



with contributions from
Ryan Martens and
Mauricio Zamora



You've got to think about big things while you're doing small things, so that all the small things go in the right direction." – *Alvin Toffler*

Abstract

Large enterprises undertake large software initiatives to be able to compete in the market. Indeed, they become increasingly dependent on software for their very business success. However the increasing complexity of the solutions they build, and their constant aging in the face of rapid technologic and business change, make the systems, and the enterprise that are beholden to them, less responsive over time. The systems become brittle, hard to maintain, and hard to evolve to meet the changing needs of the enterprise and the external marketplace. This jeopardizes future success, as the software legacy systems can no longer be quickly evolved to innovate—nor even minimally meet—the needs of the next generation of business opportunities.

The Scaled Agile Framework is designed to help enterprises address the current and future software solution challenges by providing a set of proven patterns enterprises can use to build new software assets and more rapidly evolve current systems. Core to SAFe is an understanding and emphasis on the critical importance that *software architecture* plays in helping to assure business success. In this article we reprise and revise a set of *seven principles of agile architecture* designed to help the enterprise reason about software design and software architecture, and balance the role that these two different elements play in building and revising enterprise class software systems.

[Alex Yakyma](#) is an Associate Scaled Agile Framework methodologist and agile consultant, trainer and who has been applying and advancing agile practices for over a decade. Alex fluent in English, Russian, Java, C++, TDD and ATDD, works in developing and applying the framework for multinational clients in the US, Europe and Asia.

[Dean Leffingwell](#), software industry veteran and Lean Systems Society Fellow, is a renowned methodologist, author, coach, entrepreneur and executive, having founded numerous high tech companies. He most recent books are [Agile Software Requirements](#) and [Scaling Software Agility](#). His current project is the [ScaledAgileFramework.com](#), which describes a comprehensive system for scaling Lean and Agile practices to the largest software enterprises.

Visual Studio 2013 Has Arrived



On June 26th Steve Ballmer announced the Visual Studio 2013 Go-Live preview at the Build summit. On top of the already feature rich quarterly releases this named version provides great new features and functionality we can all put to good use.

Agile Portfolio Management

A major focus with this release that provides all levels of the Enterprise visibility into their view of the project with Backlogs that now start at the highest business initiative level and provide both upward and downward “view mobility” through a reported 7 levels of hierarchy. This is part of a continuous effort to build out enterprise agile capabilities.

Version Control

There are too many changes to cover in this space so we’ll focus on a few of the top things.

There is a new visually appealing Team Explorer home page that provides easy navigation to your web based task board and the list of solutions in your workspace.

To those who yearned for the pre-VS11 style Pending Changes window, Microsoft listened and introduced “Pop-out Team Explorer pages” that with a simple click allow you to pop-out the page and dock the pending changes window anywhere you want. You can also “pop out” other pages too (like the build pages) with more planned to be added by RTM.

The Git innovations you seen recently on Team Foundation Service have been added and will ship in the on premises TFS in TFS 2013.

Coding

Coding joins Top Gun with a new “heads up display” feature in Visual Studio that provides key insights into your code as you are working. These “indicators” help you to learn more about your code and are but the first of many planned.

Memory diagnostics and memory “snapshots” enable you to find memory leaks in production. Two snapshots can be taken and compared to see what objects changed.

Testing

Visual Studio Testing capabilities continue to evolve and in VS/TFS 2013 are raised to a new level with more new capabilities. Test case management now allows you to more fully manage your test plans. You can now create/modify test plans, suites and shared steps on the web without having to switch to the Test Professional client.

With Cloud load testing you can now load test your apps without configuring any infrastructure. Use Visual Studio Ultimate Edition to create a load test and point it at Team Foundation Service and say Go! The number and size of runs available is limited until the service releases.

Release Management

Microsoft's acquisition of InRelease, a great release management solution that's been natively built to work well with TFS, allows you to manage all of your in-flight releases. For each application and each release you can define paths in an automated deployment pipeline that have stages, acceptance criteria, approvals etc. This functionality will seamlessly integrate with TFS and VS and is planned to evolve even more over time.

Team Collaboration

The always "on" Team Rooms are durable collaboration spaces that "permanently" record everything happening in your team. Notifications – checkins, builds, code reviews, etc are all configurable and reside in the Team Room as a living record of the activity in the project. A Team Room also serves as a conversation space with the rest of your team.

Conclusion

This is but a few of the highlights of the new Visual Studio 2013. There are many more to cover in the issues ahead.

Click on the banner above to download your Visual Studio 2013 Preview.

NOTE! This Preview is covered by a Go-Live license and is fully supported for production use.

Look for more information about VS 2013 in these pages in the coming issues.

The following are some more links to good information about VS 2013 Preview:

<http://www.microsoft.com/visualstudio/eng/2013-preview>

<http://blogs.msdn.com/b/bharry/archive/2013/06/03/visual-studio-2013.aspx>

<http://blogs.msdn.com/b/somasegar/archive/2013/06/26/visual-studio-2013-preview.aspx>



Lab Manager Demo



Steven Borg
Co-Founder
ALM Consultant
[Northwest Cadence](#)

Interested in using Lab Manager with SCVMM? Not sure how to test your installation? Check out the demo below! Using Standard Environments is fairly straightforward, but many people have challenges validating that they have a correct Lab Management infrastructure when using a System Center Virtual Machine Manager (SCVMM).

In this Lab Manager demo, Steven Borg will cover several components in Lab Center, how a minimal architecture is defined for a SCVMM implementation, and how to verify that all the pieces are working effectively together. This speedy demo also includes a quick walk through the basics around the Lab Manager Build Template and how to use a Team Build to create a simple environment build test.

Recommended Audiences

Business Decision Makers, Developers, Technical Decision Makers

Streaming Video only available when connected to the Internet

When Online tap this screen icon to play the video
Once playing, tap the full screen icon and rotate your tablet

Steven Borg is Co-Founder and Strategist at [Northwest Cadence](#). A Microsoft ALM MVP since 2005, Steven speaks regularly at software development and Visual Studio conferences, has authored Microsoft courseware and white papers on Team Foundation Server, Scrum and lean software development. Steven brings successful lean and agile adoption to companies currently using traditional project management techniques.



[Northwest Cadence, Inc.](#) provides training and consulting services across the United States. The exclusive focus on Software Process and Microsoft ALM tools ensures clients benefit from deep technical expertise, sound process acumen, and hundreds of practical experiences on ALM projects. They partner with software development teams to improve processes and business results.

Microsoft ALM Certification

– Part 2 of 3



Anthony Borton
ALM Trainer & Consultant
[Enhance ALM](#)

W

elcome to the second of my three part series on Microsoft's new MCSD – Application Lifecycle Management certification. In the [first part](#) of this series I provided information about the certification as well as some guidance on how to prepare for the [70-496](#) exam.



Application Lifecycle Management

In this article, we'll move on to how to prepare for the [70-497](#) exam. NOTE: You do not need to complete the exams in a specific order. I'd suggest you start with the one you're most confident on and then move through the remaining two.

Exam 70-497: Software Testing with Visual Studio 2012

This is Microsoft's first exam focusing primarily on Microsoft Test Manager (MTM) 2012. While this is the second version of MTM, it is the first time there has been an exam for it.

You can find the specific details of the exam on Microsoft's exam page for [70-497](#).

The audience identified for this exam on the exam page is "Developers". I would question this based on the *skills being measured* section of the exam page. I suggest that this exam is more directed at full time manual testers more so than developers.

Exam Details

Exam code:	70-497
Exam title:	Software Testing with Visual Studio 2012
Release date:	September 25th, 2012
Pass mark:	700
Questions:	Approximately 45
Time:	Approximately 145 minutes

	Testing for Continuous Delivery with Visual Studio 2012 (Free eBook) (Physical book on Amazon) Authors: Larry Brader, Howie Hilliker , Alan Cameron Wills
	Professional Team Foundation Server 2012 Authors: Ed Blankenship, Martin Woodward, Grant Holliday and Brian Keller.
	Professional Application Lifecycle Management with Visual Studio 2012 Authors: Mickey Gousset, Brian Keller and Martin Woodward.

That concludes part 2 of this series. In my final post in this series, I'll provide guidance on exam 70-498: Delivering Continuous Value with Visual Studio 2012 Application Lifecycle Management.

[Anthony Borton](#) is an ALM trainer and consultant for [Enhance ALM](#), an Australian based company specializing in helping organizations successfully adopt ALM best practices. He is the author of eight TFS 2012 training courses and delivers the most comprehensive range of TFS courses in the world today. You can find out more about these courses in the United States at www.quicklearn.com.

Release Management with Team Foundation Server 2012



Martin Hinshelwood
ALM Consultant
[Northwest Cadence](#)

While on the surface it looks like TFS 2012 has little in the way of support for release management, you would be wrong. There are many features in Team Foundation Server, many of them added in TFS 2010 that can aid you in creating a release management strategy.

A professional Development Team is supposed to create potentially shippable quality output that has no further work required for delivery.

There are many facets to release management and contrary to popular belief not a single one of them revolves around branching or code. There, I said it, its not about the code. Its about what you deploy. You may have a passing resemblance between code branching and your release process but they are two independent and only occasionally related processes. Intertwining them is where many organizations start to see friction, it is however avoidable.

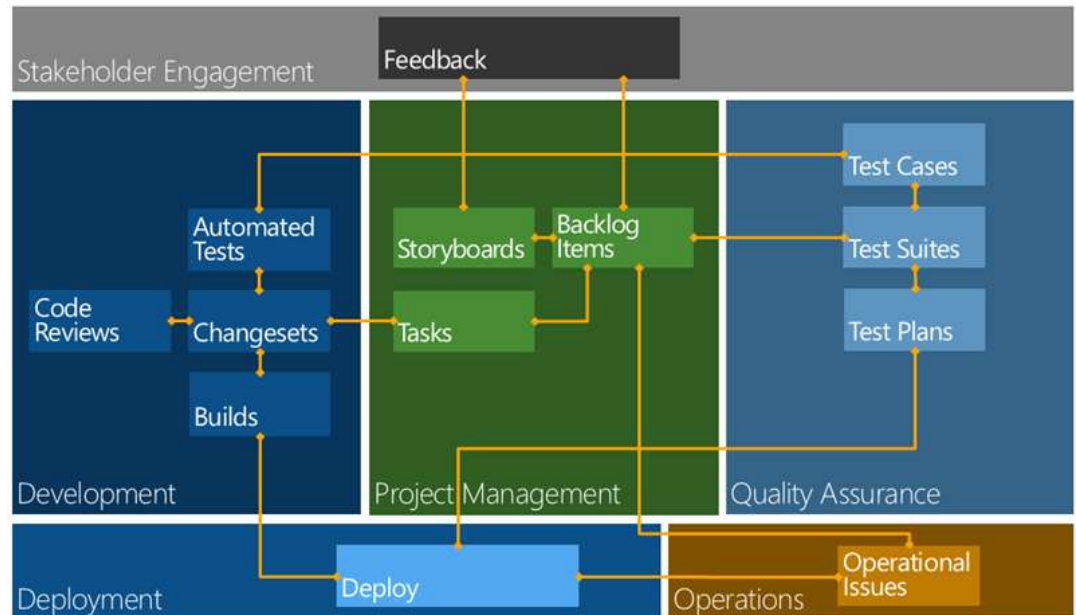


Figure: Workflow and relationships in Team Foundation Server 2012

Above I have a rough workflow for working with Team Foundation Server 2012. In this model you would create a 'build' from source control and the output would be 'dropped' to a well known location.

If you are using NuGet packages you will even know what is deployed.

Conclusion

There are lots of options for the automation of your application deployment. I can't say this enough but:

A professional Development Team is supposed to create potentially shippable quality output that has no further work required for delivery.

issues so that you don't get many, if any, issues in production. Teams that follow models like this, when combined with good engineering practices, spend less time fighting fires and more time delivering real customer value. The goal would be to get from Check-In to your Production Environment at least every 30 days.

That means that you need to be using the very best Processes, Practices and Tools to make it as slick as possible.

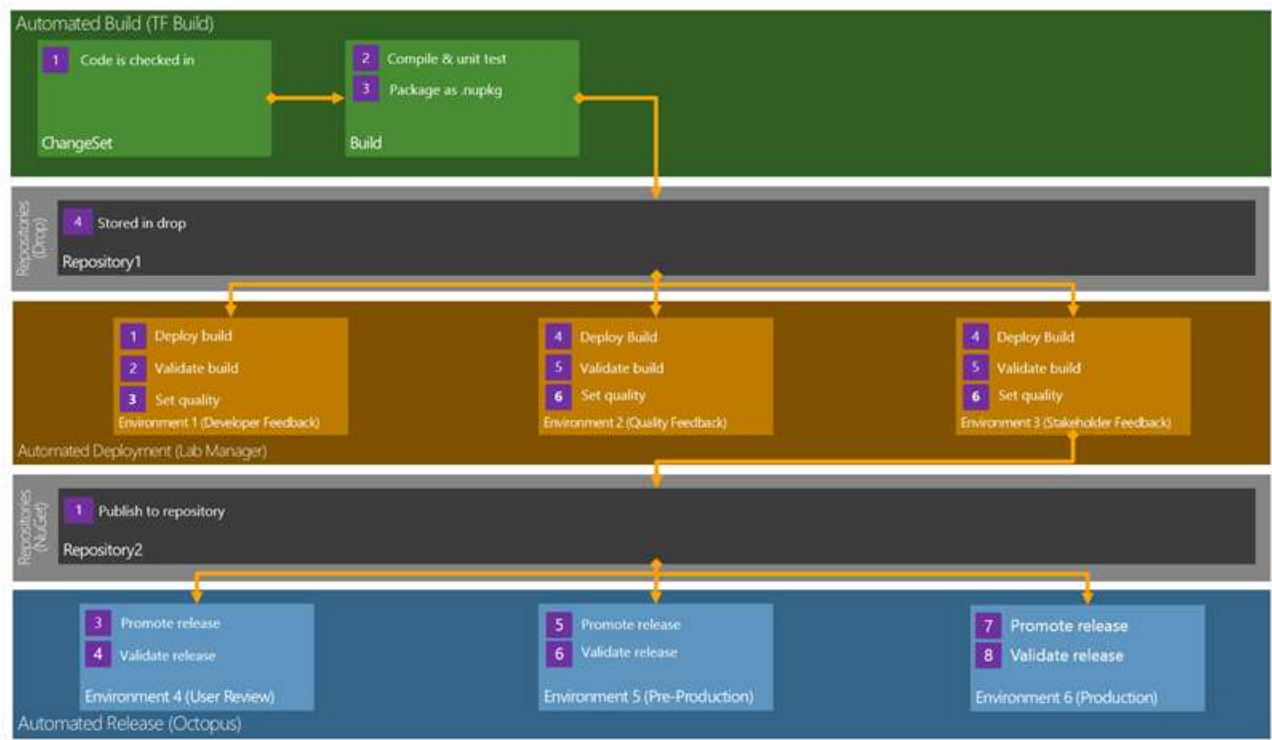


Figure: Release Management Workflow in Team Foundation Server 2012

If we fail any sort of stage gate that occurs after the engineering team has said that it is 'done' then we need to seriously look at why that happened and what we can put in place to make sure that it never happens again. That sort of attitude will improve the quality of your software and reduce the risk of delivery vastly.

This flow of building once and then repeated validation will help weed out those last quality

How long is your release process?

How sure are you of your quality?

Martin is an ALM Consultant at [Northwest Cadence](#), and [Visual Studio ALM MVP \(2013\)](#). He was named [ALM MVP of the year 2011](#), [ALM Ranger Champion for 2011](#) and was one of the first [Professional Scrum Trainers](#) with Scrum.org. Martin regularly delivers consulting and training while writing for his [Visual Studio ALM blog](#), and is a sought-after speaker.

Continuous Delivery: Delivering IT to the Always Up-to-date User



Andrew Phillips
XebiaLabs

Trying to serve today's always connected, always updated IT user with anything less than a fully automated application delivery process makes no sense. Software Delivery is evolving from an effort-intensive, error-prone "big bang" activity to a seamless, incremental and continuous process that is proving to be faster, more reliable and more efficient.

With today's always-connected user, able to access your services all the time, everywhere, and to evaluate your competitors' offerings with a few taps or clicks, the business of service delivery is never "done". Your service is expected to be always available and continuously improving.

Businesses understand that they need to be "always-on". This requires constant, consistent application and service delivery to ensure a positive, seamless, rich customer experience.

Certain software delivery and integration strategies are central to allowing your organization's offering to be reliable and engaging, meeting and exceeding the high expectations of today's audience that demands its content and applications be constantly available and instantly gratifying.

Expanding on from the widespread adoption of Agile development practices in the enterprise, the leading strategy to achieve this today is *Continuous Delivery*.

Continuous vs. Traditional Manual Delivery

In most organizations, the software delivery process today involves many manual or semi-automated steps, requiring a significant investment in terms of scripting and troubleshooting. Further, the delivery and release process followed varies widely across departments, applications and technologies, leading to error-prone, non-transferable activities and costs that increase in direct proportion to the release volume.

Continuous Delivery is a fundamental shift in the software development and delivery process, based on a simplified release management practice that unifies development, production testing, QA, testing and operations



POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with Xebialabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

Xebialabs
the deployment automation company

Discover the Xebialabs
Continuous Delivery Platform.
www.xebialabs.com

faster pace, features will land in your customers' hands much more quickly, and with a far lower failure rate. This delivers not only the latest and greatest your company has to offer, but also aligns the continuous delivery of software with today's connected culture giving rise to a seamless customer experience and greater innovation.

Making the Transition

Beginning the process of changing over to Continuous Delivery begins with an honest assessment of your existing model and your future needs. This will help you determine the best order and approach for your implementation, and give rise to realistic expectations and deadlines.

Engaging your development team is an essential element to ensuring a positive shift into automated application delivery. Not only

will their insight and experience be invaluable for understanding the challenges you have to overcome, their participation is required to accurately determine the best strategies for optimizing a new process.

The next step is to review technologies that underpin a move to Continuous Delivery, including tools for Continuous Integration, automated testing, provisioning and automated deployment. Oftentimes, organizations are finding they have some of these tools in-house already, and it is a matter of adding some final pieces to make the move to Continuous Delivery.

Once the elements are all capable of working together in order to deliver improved functionality, continuous application delivery becomes the new standard.

[Andrew Phillips](#) is Vice President, Product Management at [Xebialabs](#), a global provider of Application Release Automation offering a Continuous Delivery platform to help enterprises improve and accelerate the application delivery process. To get started moving to Continuous Delivery, [Read the new white paper](#).

What's New in TFS 2012?

– Team Explorer



Mohamed Radwan
Senior ALM Consultant
[Marvel ALM](#)

This series introduces what's new in [Visual Studio 2012](#) and [Team Foundation Server 2012](#) (TFS 2012). This article focuses new features in Team Explorer.

There are many clients for TFS (Team Foundation Server) like, Microsoft Excel, Microsoft Project, Team Web Access, and several other clients including Team Explorer.

What is Team Explorer?

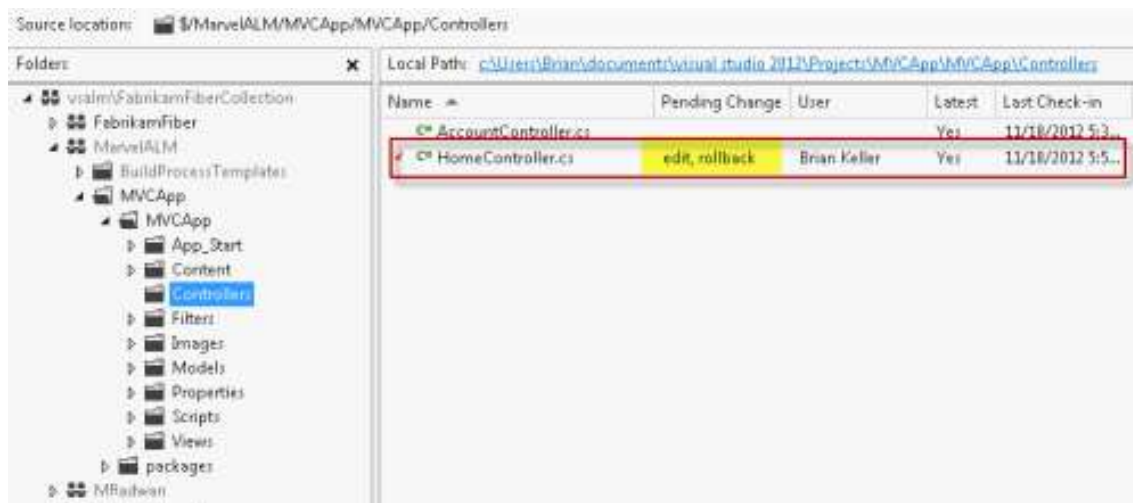
Team Explorer is considered the primary client for TFS. It is an add-in to [Visual Studio 2012](#) or it can be installed as a separate standalone application.

There are a lot of changes, new and merged features in the new Team Explorer and Visual Studio 2012 e.g. the Pending Change Window merged within Team Explorer. Also there are new windows added to the Team Explorer, e.g. there is a new window called My Work, introduced in TFS 2012.

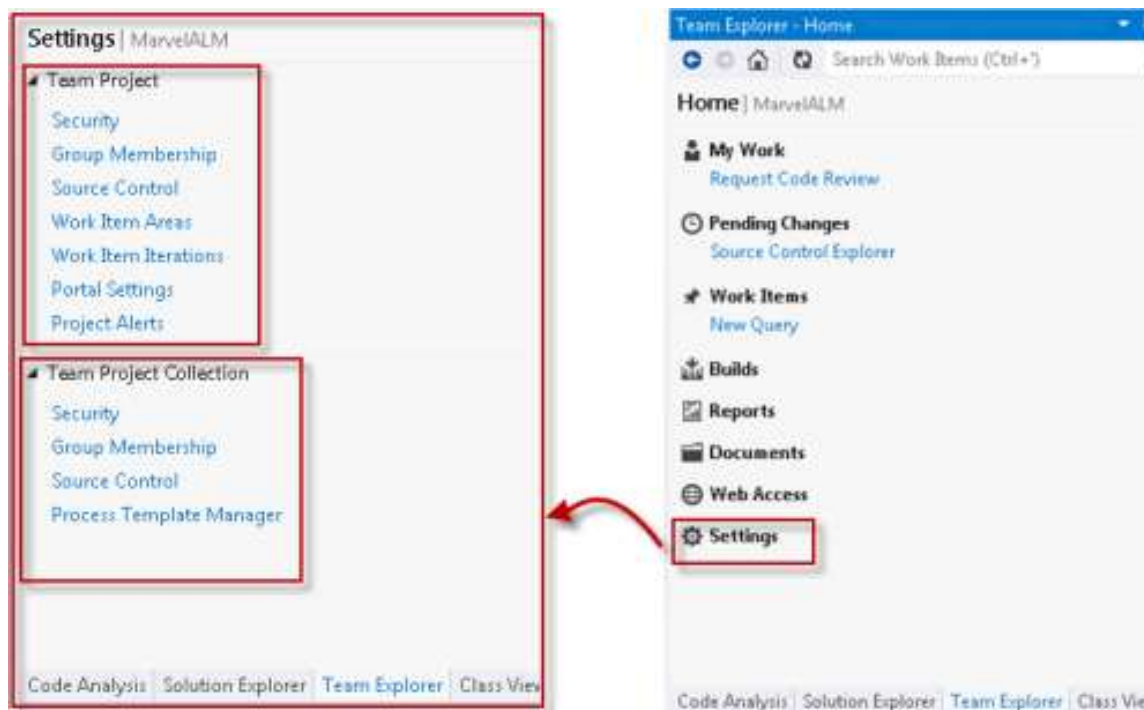
In this post I will focus on the Team Explorer itself. and its key features.

Team Explorer 2012 Enhancements

1. Architecture (Web Based)
 - a. Navigation and Extensibility
 - b. Work item search
2. New and Merged Windows
3. On Demand Data retrieval (paging)
4. Smart commands and Context driven
5. Reducing Modality
6. Performance and Async Operations
7. Rollback in the UI



It's not part of the Power Tool as a third part library anymore, it now become in the box. The project settings now also looks different.



Summary:

Team Explorer one of the major change of the TFS 2012 and there are a lot of new features and enhancements that can't be cover in one or two posts, as a summary, Team Explorer has different experience with a lot of change so this post just highlight the main and significant changes of the new experience introduced in Team Explorer.

[Mohamed Radwan](#) is a Visual Studio ALM MVP, Senior ALM Consultant at [Marvel ALM](#) with more than 10 years of software industry experience, he focuses on C#, MVC, JQuery, BDD, TDD, TFS and Practical Agile Implementation. A founder of [TFSEG](#) User Group, co-founder of MEAALM Community, author of DevMagicFake Mocking Framework, Mohamed is a frequent [blogger](#) and speaker at Microsoft events.



Empowering Rapid Delivery of Quality Software

The e-Magazine for:

ALM - Application Lifecycle Management

DevOps - Development & Operations

Agile - Scrum, Kanban, etc.

TFS - Team Foundation Server

[Click for more Details...](#)

ALM Mag Brings You Each Month:

- **Articles, Whitepapers, Case Studies and Videos and Audio** providing:
 - Advance notice and Previews of new versions,
 - **Reviews** of new versions and updates,
 - **How To and Step-by-Step Guides**,
 - **Tips** highlighting **Primary Tools** and
 - **3rd Party Integrated or Bridging Tools**,
 - **Processes, Books and Events**
- **Surveys** to discover more about the types of content that interest you so we can serve you better
- **Industry Study and Survey Results** to provide you more insight into what others in your community are doing
- **Interviews with top ALM Leaders** to help you discover what's coming and how you can get there
- **Tutorials and Training Courses** to increase your knowledge and leverage your career
- **Event and Conference Information** with a spotlight on Details, Agenda and Calendar so you are aware of what you want to attend to increase your career potential
- ...and finally **IT Humor** (yes it does exist) including **Cartoons**

and more...

This Magazine Is For You If You Are:

- a CEO, CIO, COO
- an Application or Development or QA or DBA Manager or Architect
- a Development or QA Lead
- a Product Owner
- a Project Manager
- a Scrum Master
- a Business Analyst
- a Tester
- a Developer
- ...Or anyone involved in software Planning, Development, Testing, Infrastructure Provisioning or Deployment

[Subscribe](#)

[Index Page](#)

ALM Mag.com

Empowering Rapid Delivery of Quality Software

swirly-stars-backgrounds-wallpapers-2

TFS Migrations

2008 & 2010 to 2012

Audio Podcasts

Technology & Process with the Experts

ALM Consulting Update

Industry Case Studies

Industry Surveys

**Interviews with
Industry Leaders**

Best NEW Features

**Event
Wrap Up
& Review**

**More
BONUS
Video**

TFS -
Beyond
Source
Control**

Index Page



POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with Xebialabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

Xebialabs
the deployment automation company

Discover the Xebialabs
Continuous Delivery Platform.
www.xebialabs.com

next thing. The time box limits the amount of work a team can take on. Effective teams generally use the Kanban principle of Limited WIP, only working on one or two transactional stories at a time. Some teams choose not to commit at all, but simply pull in stories according to their WIP limit, being careful not to start something that cannot be finished in a sprint. The important thing here, is not whether teams make commitments, or how they make them, but the fact these are *cross-functional* teams, collaborating with requesters, managing their own time and process, and focusing on the *completion of work*.

If organizations wanting to adopt Scrum would be willing to shift from a project culture to a team culture, they'd find many of the problems associated with project management would disappear. They would reduce (project) management overhead costs, stop calling in consultants to fix whatever the previous consultants screwed up, and they'd see a stream-lined, effective organization emerging right before their eyes. Trust the teams. It's as simple—and terrifying—as that.

[Tobias Mayer](#) is a facilitator, guide, and coach, specializing in Agile transformation. His current work explores the concept of [craftsmanship](#) in this context, focusing on the values and principles that foster an Agile mindset.

Featured Event



15

Features

- 04** ***Welcome to ALM Mag***
Editor's Welcome



- 05** ***Magic Quadrant for
Application Life Cycle Management***
The June 2012 Findings
by Thomas E. Murphy, Jim Duggan



- 07** ***A Lap Around Visual Studio 2012***
See new enhanced User Interface, Features and functionality
by James Tupper



- 08** ***Integrating Development &
Operations for the Enterprise***
The adoption of DevOps is now an established industry trend
by Keith Pleas



- 16** ***Requirement management in the modern application lifecycle***
Managing requirements can be hard... here are some answers
by Martin Hinshelwood
- 32** ***Podcast Podium***
Interviews with Agile Evangelists
by Derek Neighbors
- 33** ***TFS Taskboard on Microsoft Surface***
Goodbye Post-It Notes
by Anna Russo
- 34** ***Agile At The Enterprise Level***
Agile requires consistent collaboration among stakeholders
by Bob Savelson
- 36** ***Load Testing Using Windows Azure***
A reusable inexpensive solution with easy infrastructure management will increase performance testing adoption
by Tarun Arora
- 42** ***Team Culture, Project Culture***
Team culture offers a simple approach to solving complex problems
by Tobias Mayer
- 46** ***The History and Back Story...***
The Chicken and the Pig Story Brought to Life
by Mike Vizdos



Welcome



Keith Denham
Publisher

Treasure Cove Publishing
Volume 1 – Issue 1
January 2013

ADVISORY BOARD:

Sam Guckenheimer
Group Product Planner for
Visual Studio & Author
- Microsoft

Keith Pleas
Organizer & Conference
Chair - ALM Summit

EDITORIAL REVIEW:

Mike Vizdos
Agile Coach
Certified Scrum Trainer
Clementino de Mendonca
National ALM Architect
- Neudesic

Subscriptions:
almmag.com/subscribe

Email Address Changes:
support@almmag.com

Mission Statement:
“Our mission is to promote,
support and empower
rapid delivery of quality
software.”

Welcome to the premier issue of ALM Mag, the only online e-magazine dedicated to focus on topics of interest to the ALM, DevOps, Agile and TFS community.

This column will be an occasional addition to an edition. I much prefer the community experts take the stage maximizing the benefit through the provided value you'll get from each issue. Their contributions are presented as true to the original format as possible. You may even notice some “English” spelling.

However, I feel it important to set the stage and let you know that this magazine is to be all about the content. Content covering a wide spectrum of interest to the community: topic areas and different viewpoints.

Viewpoints will be provided in both overview and in-depth formats sometimes taking a position, outlining pros and cons, pointing out strengths and weaknesses, asking the tough questions in order to stimulate thought, not just rubber-stamping a process or tool but mostly striving to provide balanced coverage and commentary.

This electronic publication will feature the following and more:

- Articles and Videos: Reviews, How To, Step-by-Step, Top Tips
 - Primary tools, Process, 3rd Party Integrated or Bridging Tools, Books, Events
- Surveys, Studies
- Interviews – ALM Leaders, Evangelists
- Tutorials, Courseware
- Event Spotlight, Calendar
- IT Humor / Cartoons

ALM Mag the magazine will be available on desktops but primarily aimed at the three major mobile tablet platforms: pcPad, aPad and iPad.

I wish you many hours of enjoyment over the coming issues as you build a library of knowledge and resources you can turn to support you daily.

Warmest Regards,
Keith Denham

Keith Denham

*P.S. Don't forget, this is an interactive magazine, tap on the audios and videos and tap on the highlighted links to see additional related content.
Tap on the Index Page Button at the foot of any page to return to the Index.*

Gartner Magic Quadrant for Application Life Cycle Management

Gartner.

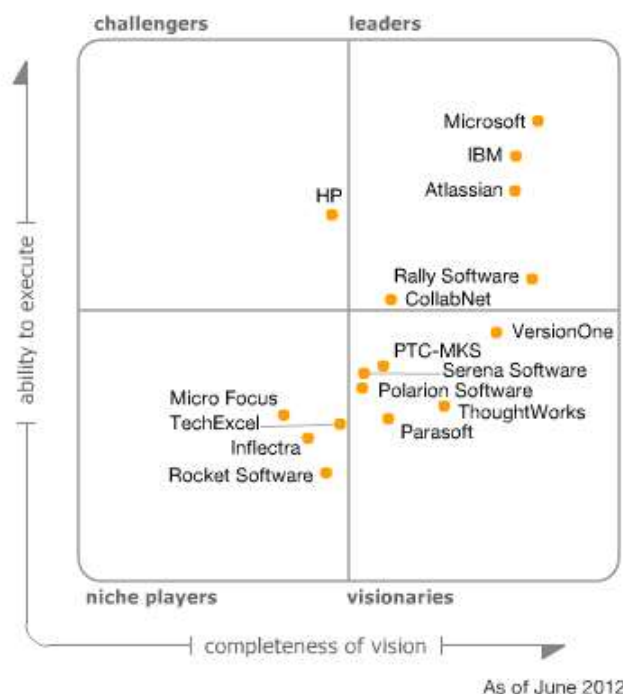
Gartner, a leading industry analyst firm, evaluated 16 vendors based on their completeness of vision and ability to execute in the Magic Quadrant for Application Lifecycle Management. We invite you to [download the complete report](#) for insights into how Agile and the cloud technologies are driving adoption of ALM.

Summarized by
Keith Denham

**Derived from the
original work by
Analysts:**
Thomas E. Murphy,
Jim Duggan
5 June 2012

The application life cycle management (ALM) tool market is focused on the planning and governance activities of the software development life cycle (SDLC). Traditionally, this has been the combination of software change and configuration management (SCCM), requirements management and quality management. ALM is evolving, with a focus on coordinating the planning and execution of software projects with an emphasis on work items, planning iterations and coordinating actions

Figure 1. Magic Quadrant for Application Life Cycle Management



**Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.*

A Lap Around Visual Studio 2012



James Tupper
ALM Consultant
[Northwest Cadence](#)

View this introduction to the major new features and improvements in Visual Studio 2012. Expect to see the new enhanced User Interface, Agile Planning Tools, Requirements Gathering Tool, Stakeholder Feedback Tool, Updates to the Developer and Tester Experience, Version Control Improvements, and DevOps Integration.

The list goes on but the ride starts here, so buckle up and join us for this lap around Visual Studio 2012.

Recommended Audiences

Business Decision Makers, Developers, Technical Decision Makers

Streaming Video only available when connected to the Internet

When Online tap this screen icon to play the video
Once playing, tap the full screen icon and rotate your tablet



Click on the graphic above to play the video

If the graphic above doesn't play for you

[click here for a dedicated player and the option to Download](#)

James Tupper, ALM Consultant at [Northwest Cadence](#), is experienced with both software and hardware development, extending Visual Studio Team Foundation Server using its APIs, and customizing TFS in unconventional ways. James has also developed solutions using the TFS work item tracking for complex scenarios not centered on software development. A certified Professional Scrum Developer, James writes often on the Northwest Cadence blog.



[Northwest Cadence, Inc.](#) provides training and consulting services across the United States. The exclusive focus on Software Process and Microsoft ALM tools ensures clients benefit from deep technical expertise, sound process acumen, and hundreds of practical experiences on ALM projects. They partner with software development teams to improve processes and business results.

Integrating Development & Operations for the Enterprise



Keith Pleas
Organizer and
Conference Chair for the
[ALM Summit](#)

Executive Overview

Organizations that use practices and tools to integrate development and operations can significantly speed up their ALM process and enable continuous delivery of software. With continuous delivery, internal and external customers benefit from quicker access to new applications, features, and fixes – and the organization benefits from that increased usage and satisfaction as well as having higher quality software at a lower cost. The philosophy and practice of this deeper integration between development teams and operations is often referred to as DevOps.

DevOps practices are particularly well suited to the *Software as a Service* (SaaS) delivery model used in cloud computing, where continuous delivery has become standard. These successful practices are being extended to the enterprise to realize the value of continuous delivery. This consistency is even more important for companies whose applications bridge the gap between private and public clouds through *hybrid IT*¹. Many forward leaning organizations are further able to fuse these two traditionally distinct roles into one single DevOps role, resulting in even less friction throughout the ALM process.

The adoption of DevOps is now an established industry trend supported by numerous vendors, including many ALM tool providers. Looking forward, analysts at Gartner recently predicted that “By 2016, 40% of organizations will implement DevOps processes to support continuous delivery and simplified release management.”² Visual Studio 2012 can be

Keith Pleas is the Organizer and Conference Chair for the [ALM Summit](#). Keith is also a software architect, and has worked with the Microsoft [patterns & practices](#), .NET Framework, and Visual Studio teams. Keith is an internationally known writer, speaker, conference organizer. Keith was a founding board member of INETA and created the [INETA Speakers Bureau](#).

¹ *hybrid IT* enables companies to use private clouds for application delivery and also take advantage of public cloud infrastructure to create resource flexibility and address temporary infrastructure demand needs

² Gartner Predicts 2012: Application Development (Dec 2011)

sponsored by Microsoft

ALM Summit 3

expertise worth sharing

January 29-31, 2013 - Microsoft Conference Center, Redmond

Register at www.alm-summit.com

follow us on



keynote speakers



Dean Leffingwell

Creator, Scaled Agile Framework

"Being Agile. Scaling Up. Staying Lean"

Scott Porad

CTO, Cheezburger Network

"Lean Startups, Cheezburgers, and You"

Brian Harry

Technical Fellow, Microsoft

"Building an Engineering Organization for Continuous Delivery"

Jason Zander

Corporate Vice President, Microsoft

"Continuous Services and Connected Devices"

Sam Guckenheimer

Product Owner, Microsoft

"Reimagining the Application Lifecycle"

James Whittaker

Testing Guru, Microsoft

"Re-inventing the Internet"

conference tracks

ALM Leadership • Agile Development • DevOps • Testing

sponsors



Requirement management in the modern application lifecycle



Martin Hinshelwood
ALM Consultant
[Northwest Cadence](#)

Managing requirements is hard and no single tool can tell you how to achieve that within your organization. To that end, Visual Studio 2012 Team Foundation Server stays out of the “where do your requirements come from” world and firmly in the “I have my requirements.. what now?” world and indeed the tools built on top of the requirement management system stay in one niche.

UPDATED 2012-09-11: I have updated the list of tools available

The Modern App Lifecycle

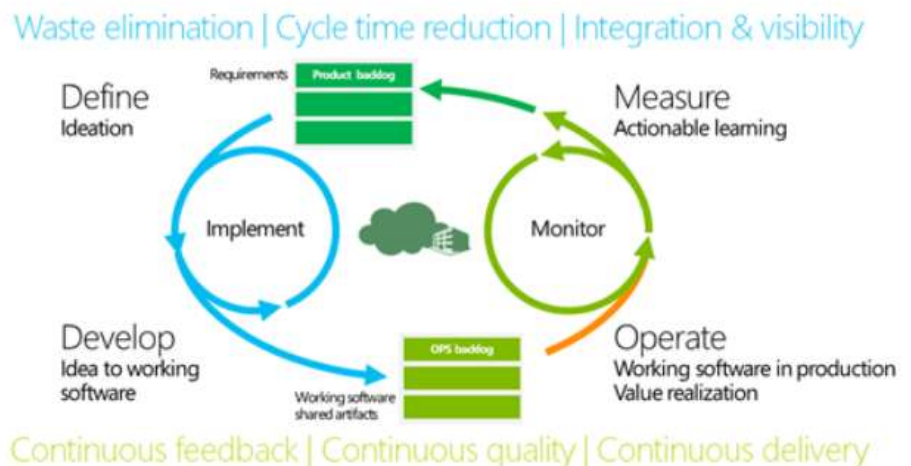


Figure: Requirement Management is part of Define / Ideation

Team Foundation Server aims to become your matter of record or single-point-of-truth for what state your requirements are in, what the details of that requirement are, who is currently working on it and who changed what when.

The system has lists of work item that can have any number of fields with configurable forms and permissions along with many other features out-of-the-box and you can store data against these fields with TFS storing a

Martin is an ALM Consultant at [Northwest Cadence](#), and [Visual Studio ALM MVP \(2013\)](#). He was named [ALM MVP of the year 2011](#), [ALM Ranger Champion for 2011](#) and was one of the first [Professional Scrum Trainers](#) with Scrum.org. Martin regularly delivers consulting and training while writing for his [Visual Studio ALM blog](#), and is a sought-after speaker.

Modern Requirements Management



inteGREAT & TFS share a single database

Key Features that extend TFS capabilities

Adaptive project methodology support	Completely configurable taxonomy	Project baselining	Automatic document generation
Automatic test case generation	Reusability across projects	Impact assessment	Author and review in WORD
Simulations and on demand Wireframes	Non-Functional requirement library	Automated Function Point estimation	Specification Checker to validate requirements
Wizard authoring for completeness	Legacy Word & Excel import	Conceptual data model creation	Drag & drop traceability

Interviews with Agile Evangelists



Derek Neighbors
[Integrum Technologies](#)

Agile Weekly PodCast

[Agile Manifesto with James Grenning](#)



*Click on the graphic above to play the interview
If the graphic above doesn't play for you*

[click here for a dedicated player and the option to Download](#)

Derek Neighbors and James Grenning discuss:

- History of The Agile Manifesto
- Embedded TDD
- Estimates

[Agile Weekly Podcasts](#)

[Derek Neighbors](#) is a serial entrepreneur who helps people bring ideas to reality. Derek co-founded [Gangplank](#), a collaborative workspace, in 2008 to encourage local “creatives” to explore innovative ideas. A partner at [Integrum Technologies](#), a consulting services firm, helping companies build high performing teams, Derek teaches entrepreneurship at [Arizona State University](#).

TFS Taskboard on Microsoft Surface - Goodbye Post-It Notes



Anna Russo
ALM Consultant
[Imaginet](#)

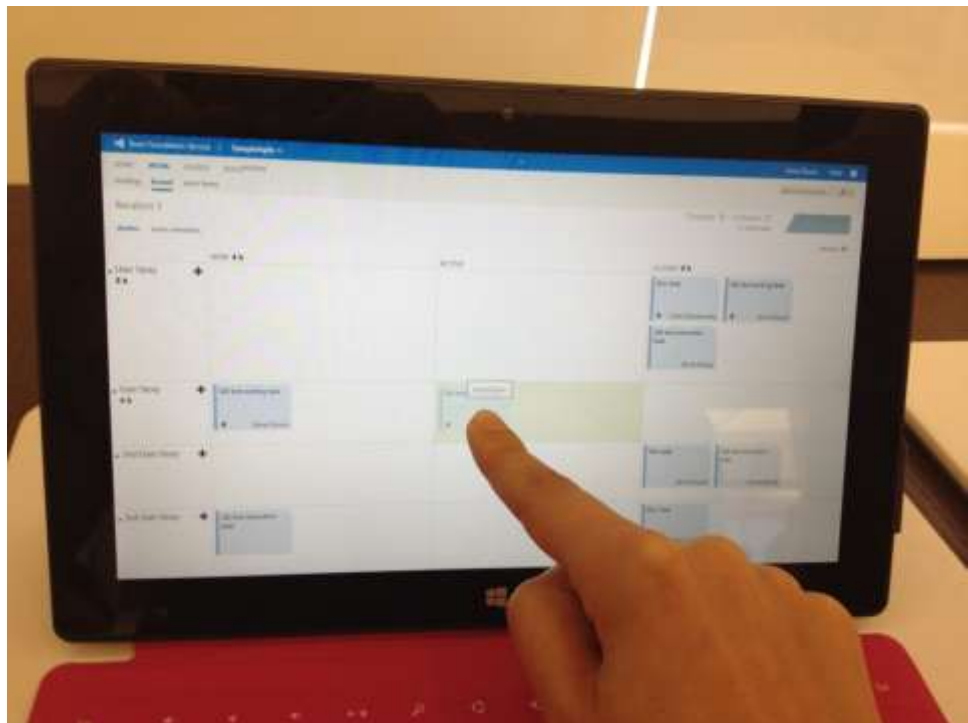
With the Microsoft Surface out I decided to give the drag and drop features of the TFS taskboard a whirl. What's great about this is that you can use the Surface for your stand-up meetings and connect it to a projector via [VGA or HDMI\(sold separately\)](#). Now that's what I call a high tech stand-up meeting!

Once connected to TFS 2012...

Dragging tasks from New to Active on the TFS taskboard is fun and easy. It's so awesome that I can update my work item status with the swipe of a finger.

It's neat that TFS has such a portable rich client and the data stored in a database where you can collaboratively work together.

Try to do that with your post-it notes :)



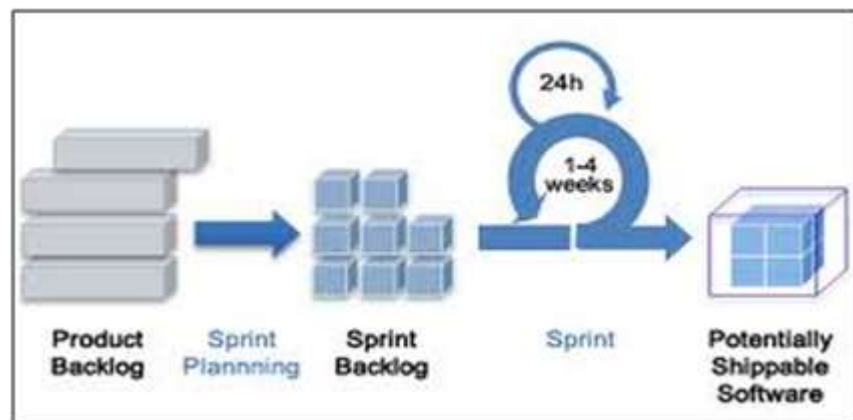
[Anna Russo](#) is an ALM Consultant and MVP with [Imaginet](#), a firm that specializes in helping companies improve software development process with TFS. She has worked with a variety of clients and upgraded/installed TFS, provided training, created custom processes, implemented release strategies, and automated build scripts. Visit Anna's blog [ImprovingSoftwareQuality.com](#)

Agile At The Enterprise Level



Bob Savelson
SVP Sales & Marketing
[eDev Technologies](#)

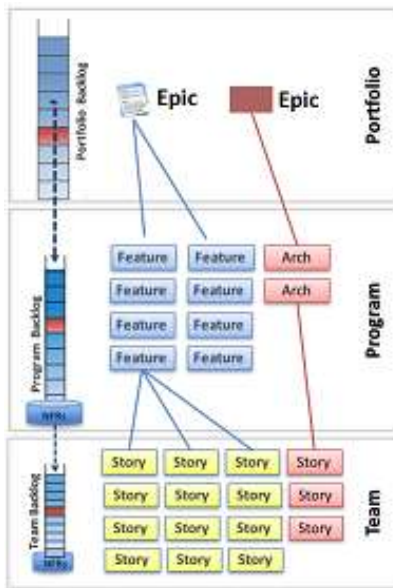
Nowadays, Agile is considered to be one of the best process models for application development in small and medium sized teams, for its incremental and iterative approach. As it is adaptive in nature, Agile requires consistent collaboration among stakeholders for better fulfillment of customer needs. Agile adapts itself to any change and maintains progress. Agile involves frequent releases,



once every few weeks, with small sets of deliverables defined in Sprint Backlogs. The focus of the Agile methodology is obtaining a small set of features to deliver business value early, and continually improving it, adding further functionality throughout the life of the project.

For its adaptive nature with constant collaboration needs and frequent iterations, Agile works fine when it comes to working within a small or medium sized team. But when applying Agile at an Enterprise-level, a number of issues need to be resolved for effective Agile implementation. For many enterprises considering Agile approaches, the absence of effective and scalable Agile requirements processes has been a showstopper for adoption.

Dean Leffingwell, Scaled lean-agile author, methodologist and consultant, has defined an excellent Scaled Agile framework for Agile adoption at an enterprise level, by eliminating issues faced by widely distributed teams and multiple parallel backlogs & iterations. In the framework, he has defined an enterprise as a group of teams working collaboratively on a set of features with effective collaboration, providing high quality applications and better customer satisfaction.

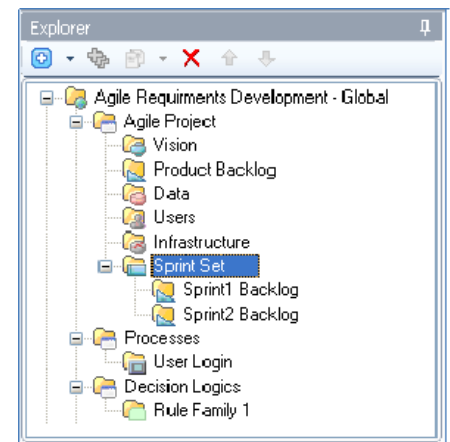


In the Scaled Agile Framework, a Product Backlog is considered as an Enterprise Backlog Model, translating the allocation of strategic investments to the portfolio, program, and team levels. The Enterprise Backlog breaks down into three levels: Portfolio Backlog, Program Backlog and Team Backlog.

The Portfolio Backlog consists of Business and Architectural Epics that are large-scale development initiatives defining the desired solutions. Epics are broken down into Features at the Program level and are maintained in the Program Backlogs. Each feature defines a set of User Stories to be implemented at the Team level. User Stories are then stored in the Team Backlog. Features and User Stories define functional requirements required to achieve an Epic, whereas Non-functional requirements are also identified and maintained at the Program and Team levels.

Leffingwell has scaled iterations into multiple levels in his Scaled Agile Framework. At the Team level, two-week sprints are set that require teams to complete their designated functionalities within the iteration. Outputs from multiple iterations are synchronized and integrated together to release the desired solutions in 6 to 8 weeks. During all these iterations and releases, stakeholders continuously collaborate with each other, and changes are absorbed as done within traditional Agile processes.

In inteGREAT, a flexible framework is provided to support Agile Requirements Development and Management. inteGREAT allows Agile teams to define User Stories through Requirements knowledge objects and maintain them in Product Backlog, further assigning them to different sprints. As inteGREAT bi-directionally integrates with Team Foundation Server, Requirement management capabilities of TFS are leveraged for managing and keeping track of requirements across multiple teams providing support to Enterprise-level Agile implementation.



[Bob Savelson](#), a seasoned veteran of the software development industry, has worked for large outsourcers and built his own requirements facilitation company; helping many clients to optimize their ALM processes and define the 'right' requirements. Today, as SVP at [eDevTech](#), he continues to promote [inteGREAT](#) and [SmartOffice](#) as industry leading requirements tools.

Visual Studio Load Testing using Windows Azure



*Tarun Arora
Microsoft MVP
ALM Ranger
Consultant
Avanade*

In my opinion the biggest adoption barrier in performance testing on smaller projects is not the tooling but the high infrastructure and administration cost that comes with this phase of testing. Only if a reusable solution was possible and infrastructure management wasn't as expensive, adoption would certainly spike. It certainly is possible if you bring Visual Studio and Windows Azure into the equation.

It is possible to run your test rig in the cloud without getting tangled in SCVMM or Lab Management. All you need is an active Azure subscription, Windows Azure endpoint enabled developer workstation running visual studio ultimate on premise, windows azure endpoint enabled worker roles on azure compute instances set up to run as test controllers and test agents. My test rig is running SQL server 2012 and Visual Studio 2012 RC agents. The beauty is that the solution is reusable, you can open the azure project, change the subscription and certificate, click publish and ***BOOM*** in less than 15 minutes you could have your own test rig running in the cloud.

In this blog post I intend to show you how you can use the power of Windows Azure to effectively abstract the administration cost of infrastructure management and lower the total cost of Load & Performance Testing. As a bonus, I will share a reusable solution that you can use to automate test rig creation for both VS 2010 agents as well as VS 2012 agents.

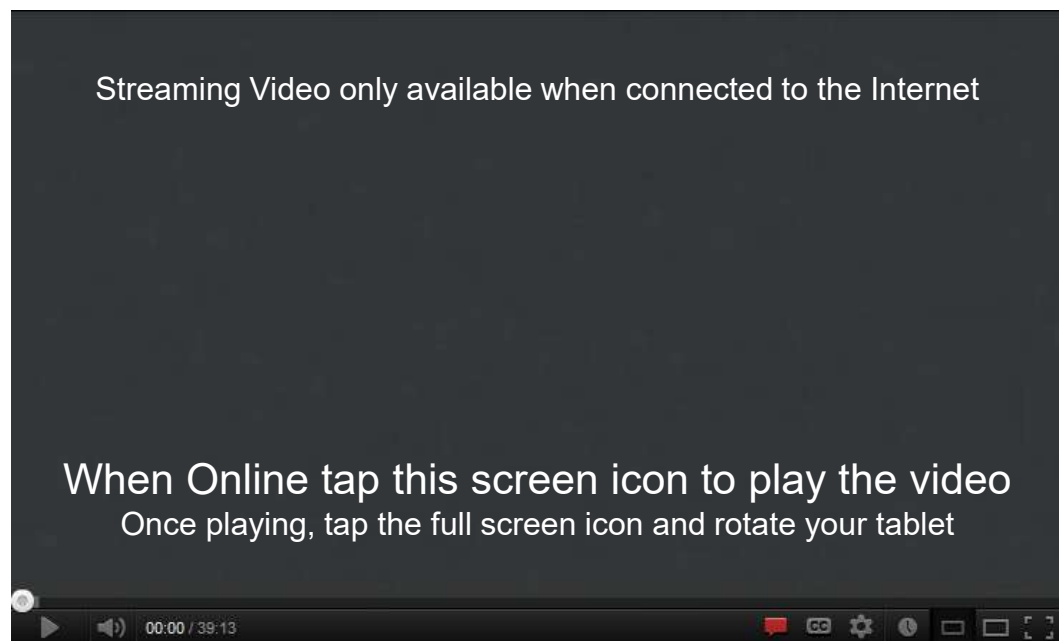
Introduction

The slide show below should help you under the high level details of what we are trying to archive...

[Leveraging Azure for Performance Testing](#)

Walkthrough

To set up the test rig in the cloud, the test controller, test agent and SQL express installers need to be available when the worker role set up starts, the easiest and most efficient way is to pre upload the required software into Windows Azure Blob storage. SQL express, test controller and test agent expose various switches which we can take advantage of including the quiet install switch. Once all the 3 have been installed the test controller needs to be registered with the test agents and the SQL database needs to be associated to the test controller. By enabling Windows Azure connect on the machines in the cloud and the developer workstation on premise we successfully create a virtual network amongst the machines enabling 2-way communication. All of the above can be done programmatically, let's see step by step how...



Video Walkthrough–Leveraging Windows Azure for performance Testing

Click on the graphic above to play the video

If the graphic above doesn't play for you

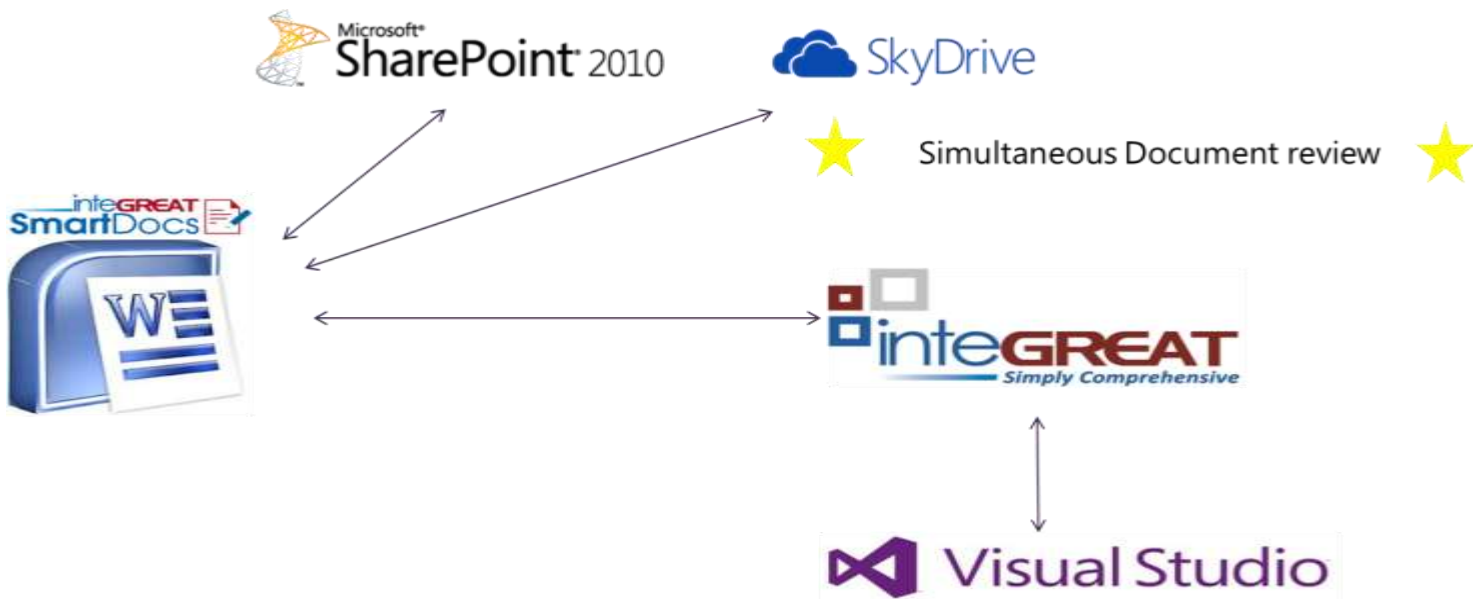
[click here for a dedicated player and the option to Download](#)

Solution

If you are still reading and are interested in the solution, drop me an [email](#) with your windows live id. I'll add you to my TFS preview project which has a re-usable solution for both VS 2010 and VS 2012 test rigs as well as guidance and demo performance tests.

[Tarun Arora](#) is a passionate technologist at [Avanade](#), London. Tarun is a Microsoft MVP and a Visual Studio ALM Ranger. After work he is found moderating MSDN forums, [blogging](#) and writing for MSDN, TechNet, Codeproject and other technology centric websites. For the latest and greatest follow him on [@arora_tarun](#)

Simplicity - Word with real time requirements repository connectivity



"inteGREAT Smart Docs provides TFS users with a very familiar user interface that will enhance its speed of adoption, while broadening its use throughout the enterprise"
Brian Harry - Product Unit Manager for Team Foundation, Microsoft Corporation.

The use of Word, Excel and Visio, while very familiar to requirements project stakeholders, lacks efficiency in reuse, change synchronization and stakeholder collaboration. inteGREAT SmartDocs solves this problem while maintaining the familiar authoring environment of Word. SmartDocs is a **Word plugin** that enables the requirements contained in these documents to be saved to a single source requirements repository; reducing cycle time through the documents' lifespan, dramatically.



Copyright eDev Technologies 2012 | www.edevtech.com

Business Users

Create a SmartDoc in Word - Upload to inteGREAT

Open a SmartDoc Word template and create requirement content. Load the document into inteGREAT and see the requirement hierarchy and traceability established.

Benefit: Any project stakeholder may contribute requirement content in Word, without having to learn a new interface.

Edit a SmartDoc in Word - Upload to inteGREAT

Open a SmartDoc Word document with existing requirements and edit the document adding new content and editing existing content. Load the document into inteGREAT and see the additions and updates reflected.

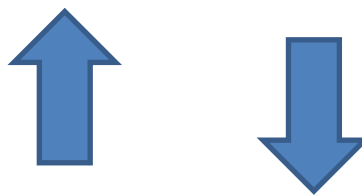
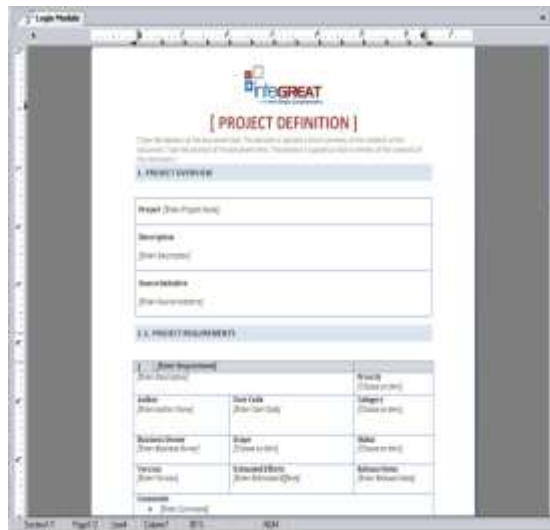
Benefit: Once a SmartDoc exists and has been used for specific requirements, any user may update this document, which in turn updates inteGREAT

inteGREAT SmartDoc Generation

Based on the requirement content in inteGREAT, have inteGREAT generate a SmartDoc, which can then be given to multiple stakeholders simultaneously. The stakeholders may update the document with track changes; return their SmartDoc to the BA who accepts/rejects changes with the results then uploaded into inteGREAT.

Benefit: Collaboration is now simplified through the use of SmartDoc Word documents. Business stakeholders can review and update SmartDocs in the familiar Word interface, and have their changes reflected in the final documentation

SmartDocs Usage Scenarios



Single source Requirements Repository



Copyright eDev Technologies 2012 | www.edevtech.com

Technical Users

Get Work Item from TFS to include in a SmartDoc

When working in a SmartDoc a user may access the existing Work Items in TFS and include them in the SmartDoc.

Benefit: Any stakeholder may reuse existing requirements, allowing for consistency across the project.

Create new Work Item in a SmartDoc - upload to TFS

New requirements created in a SmartDoc can be saved directly as work items in TFS.

Benefit: All stakeholders can contribute directly to the single source requirements repository

Update a Work Item in a SmartDoc - upload to TFS

Existing requirements which are updated by a user in a SmartDoc can be saved in TFS.

Benefit: Stakeholders contribute changes directly to the TFS single source repository.

For More information and to see SmartDocs in action please visit www.edevtech.com or call us at 1-877-473-2881

Team Culture, Project Culture



Tobias Mayer
Agile Coach

I've heard a lot over the years about how Scrum is inadequate for managing large projects—some would say managing projects at all—and much has been written about the need for Agile Project Management, or the implementation of a Lean or Kanban approach to mitigate this apparent inadequacy. Broken Scrum implementations are (we hear) fixed by the addition of one or more of these approaches, or even in combining Scrum with a defined (e.g. waterfall) approach. I don't doubt the case studies. In fact, I agree that Scrum is a half-assed way to manage projects.

But what's interesting here is that we stay stuck in the idea that projects must be managed—or even that there need be projects at all. While we hold on to our project-centric mindset Scrum will never be an appropriate solution—and I'd argue that none of the other suggested alternatives or enhancements will be much better. While we try to add layers of agility onto what is essentially an incompatible foundation we will only ever see limited success—and quick reversion back to status quo at the slightest panic. For any truly Agile effort to work we need a complete paradigm shift in how we manage business.

Back in the 1970's or perhaps earlier, the software world adopted what we might call a project culture. This means that the project is king, and the people are there to serve the project. Projects have managers, and if we add more managers we can add more projects, and get more things done at once. So goes the thinking. And what follows is that people become “resources” to the projects. It is the individual contributor who is sought after by the competing project managers, whose attention is frequently tugged in multiple directions, and teams—such as they are in a project culture—are weak entities, frequently broken apart as priorities change and skill-sets are needed elsewhere.

The project culture undermines a key principle of effective workmanship: that of focus. While individuals are pushed from project to project a massive overhead of context switching and thus loss of productivity is incurred. In simple experiments [\[ref\]](#), this cost runs around 30-50%. I'm sure it is much higher in complex environments. Because of this mess we have to hire more managers to better “level the resources”. Moving people around, assigning work, tracking that work, all this needs careful and detailed management, and good project managers are not cheap.



POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with Xebialabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

Xebialabs
the deployment automation company

Discover the Xebialabs
Continuous Delivery Platform.
www.xebialabs.com

next thing. The time box limits the amount of work a team can take on. Effective teams generally use the Kanban principle of Limited WIP, only working on one or two transactional stories at a time. Some teams choose not to commit at all, but simply pull in stories according to their WIP limit, being careful not to start something that cannot be finished in a sprint. The important thing here, is not whether teams make commitments, or how they make them, but the fact these are *cross-functional* teams, collaborating with requesters, managing their own time and process, and focusing on the *completion of work*.

If organizations wanting to adopt Scrum would be willing to shift from a project culture to a team culture, they'd find many of the problems associated with project management would disappear. They would reduce (project) management overhead costs, stop calling in consultants to fix whatever the previous consultants screwed up, and they'd see a stream-lined, effective organization emerging right before their eyes. Trust the teams. It's as simple—and terrifying—as that.

[Tobias Mayer](#) is a facilitator, guide, and coach, specializing in Agile transformation. His current work explores the concept of [craftsmanship](#) in this context, focusing on the values and principles that foster an Agile mindset.

Lean Software Development in a Nutshell



Al Shalloway
Founder & CEO
[Net Objectives](#)

Much of the work done in software development groups (IT or product) is the result of delays and interruptions. This sentence sounds funny, doesn't it? But on reflection, let's see where a lot of our time (work) is spent:

- Reworking requirements
- Building software to satisfy missed or misunderstood requirements
- Finding bugs
- Thrashing while trying to integrate two components that were built semi-independently of each other

What causes most of this work? I'd suggest delays. In the above cases, respectively:

- Delays in the time from getting requirements until using them
- Delays caused by having to do big batches of work until getting feedback on it
- The delay from writing the bug until it is discovered (fixing is fast if this delay is short, finding gets exponentially longer the longer this delay is)
- The delay from testing the synchronization of the two components

Not to mention inefficiencies create by decisions made at the wrong time (too early means a delay from when the decision was made until it should have been made, too late means a delay from when it should have been made to when it was made).

One hears a lot about batch size in lean because batches of work are the easiest things to spot. But it is actually the delays and interruptions we are trying to eliminate when we manage flow. Managing batches does this because the underlying cause of batches (too much work in progress – WIP) is the same underlying cause of delays and interruptions.

Lean's mantra of "delivering fast" is often misunderstood to mean "go fast." It doesn't mean that. It means "avoid delays and interruptions in order to get there (delivery) fast." You avoid delays and interruptions by attending to flow.

Push vs. Pull

The essence of effective Lean-Agile development is to deliver the most important capabilities in the timeliest manner. To enable developer groups to respond quickly, they must work without slowing themselves down with delays and interruptions. This means they need to accept work at their own pace. This approach is called 'pull' because they pull work from a backlog of prioritized work. This enables a more stable rate of productivity. This, in turn, creates greater predictability of the rate at which they accomplish their work. On the other hand, when work is given to them with set schedules, the set schedule typically requires more things being worked on than should be. This contributes to delays and interruptions causing additional work. This instability in the development process causes unpredictability of what they can accomplish.

Not the Whole Story

This, of course, is not the whole story of Lean. Lean provides insights into product prioritization, management, and learning.

Summary

Much of the work done in software development is not value added work but is work caused by delays and interruptions. Working from old requirements, re-working requirements, the time to find bugs, thrashing due to integration, implementing misunderstood requirements often account for more than half of the work done by a development team. Delays in, and interruptions to, our workflow is mostly due to working on too many, too large projects. We must start at the beginning of the value stream by doing proper product portfolio management. We must create visibility throughout the value stream so we can improve the efficiency of the work being done. We must manage the workflow across teams when more than one team is involved. We must shift from focusing on isolated optimizations of efficiency to a holistic view of the entire workflow.

Cause of Delay

- Big Planning Cycles
- Big projects
- Implemented in big Batches
- Specialized team members
- Validation at end

Cause of Efficiency

- Small planning cycles
- MMFs
- Incremental development
- Cross-functional, load-balanced
- Quick validation

Table 1. Causes of Delay and Actions to Counter Them

Al Shalloway is the founder and CEO of [Net Objectives](#). With over 40 years of experience, Al is an industry thought leader in Lean, Kanban, product portfolio management, Scrum and agile design. He helps companies transition to Lean and Agile methods enterprise-wide with consulting and training. Al is a [SAFe Program Consultant](#) the co-author of [four books on Lean, Scrum, Design Patterns and Agile Design](#).

ALM State of the Industry Reports: which one is right?



Clementino de Mendonca
National ALM Architect
[Neudesic](#)

Over the last 6 months I have been asked which one is better: TFS, based on the [latest Gartner report](#), or Team Concert and Rally, based on the latest Forrester report (which you get [here](#) and [here](#))? The answer is quite simple if you actually read both. Here is the short version:

Forrester's Wave report is made every two years, and it was comparing the latest version of Rally and Team Concert with Visual Studio/TFS 2010, so naturally TFS is in the second tier as the Agile backlog and planning tools were not considered;

Figure 3 Forrester Wave™: Application Life Cycle Management Tools, Q4 2012





POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with Xebialabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

Xebialabs
the deployment automation company

Discover the Xebialabs
Continuous Delivery Platform.
www.xebialabs.com

What I like about this report is its Ovum SLM Solution model, which highlights new trends such as [DevOps](#) and the growing overlap of ALM and PLM (Product Lifecycle Management). It also includes a list of vendor profiles such as [Seapine](#) and [Tasktop](#), and their current ALM products. This is very useful if you are trying to understand industry trends as opposed to tracking a single company.

Notice how the same company/ALM suite can be in a completely different ranking depending on the report focus. That's why I recommend paying close attention to the methodology used by each report vendor. The best way to understand the complexity of the ALM industry is to create your own composite view based on many sources. At a minimum you will need these four reports.

Just reading the marketing perspective from each company, which obviously will leverage those reports to provide a biased perspective of they own products, will put you in a position of an uninformed bystander in the ALM world.

Clementino de Mendonça is a Professional Scrum Developer Trainer, a [Neudesic](#) National ALM Architect, a former National Lead for ALM and Testing with Capgemini/Sogeti, and was on the Microsoft Services National ALM Team for 4 years. On the Visual Studio Process Team in 2007, he helped design the Visual Studio 2010 process templates (MSF Agile 5.0 and MSF CMMI 5.0).

EclipseCon Review



*Keith Pleas
Organizer and
Conference Chair for the
[ALM Summit](#)*

The [ALM Connect](#) conference in March, 2013 – co-located in Boston with EclipseCon – presented a variety of ALM sessions covering topics from Scrum to testing, and focusing on many of the Open Source tools that make up the typical Eclipse-based portfolio.

Lifecycle Integration

Several sessions also covered integration techniques, including “Lifecycle Integration – The secret sauce of ALM success” by ALM industry veteran Dave West, now with [Tasktop](#). Dave’s session described how the common focus on vendor-based tool suites has removed focus from the end-to-end business process, and how the tool silos are becoming more entrenched as developers and teams are empowered to select the tools that make them most productive.

A recent survey of 280 customers by Forrester Research asking “Which would you consider as major roadblocks of ALM solutions?” showed that integration has become the #1 problem in ALM:

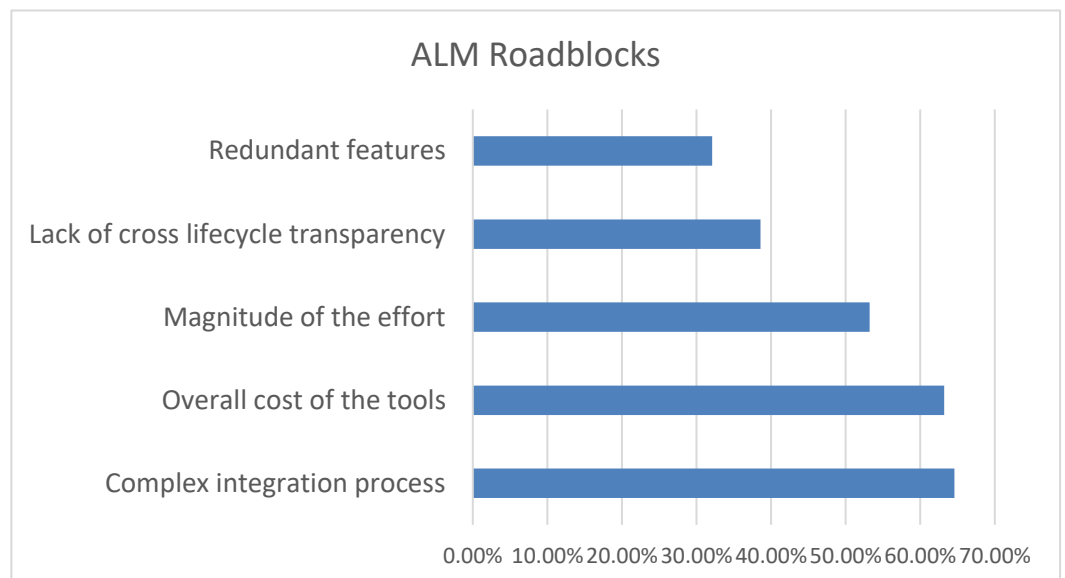


Figure 1: Reported ALM Roadblocks

Level	Focus	Characteristics	Results
5	Hypothesis-driven delivery	Requirements include testable metrics Frequent use of A/B testing Services designed for CD DBMS changed decoupled from system changes	Delivery enables business innovation
4	Release on demand	Teams organized around services Deployment pipeline rejects bad changes Work delivered in small batches Comprehensive test + release automation	Service always in a releasable state Capability >= Need
3	Regular releases w/ milestones	CI and trunk-based development Automating provisioning and testing “Done” = tested and deployed	Regular release cadence Capability < Need
2	Time-boxed releases	Clear product ownership Change management controls <1 mo. cycles Some testing, release automation	Planned releases Capability < Need
1	Heroic individuals	Manual testing Integration explosion Manual provisioning	Ad-hoc releases

Figure 6: Continuous Delivery Maturity Model

Summary

Like the old adage that the only certainty is “death and taxes”, ALM Connect and its Executive Day counterpart made clear that – while there many techniques and tools available to the development community – the forces that face everyone include “complexity and speed”. The most successful businesses will be those that best manage these many options to find the optimum combination of practices and tools that allow them to build – before their competition - the software that their users want.

Keith Pleas is the Organizer and Conference Chair for the [ALM Summit](#). Keith is also a software architect, and has worked with the Microsoft [patterns & practices](#), .NET Framework, and Visual Studio teams. Keith is an internationally known writer, speaker, conference organizer. Keith was a founding board member of INETA and created the [INETA Speakers Bureau](#).

TFS Admin Reports delivered to your inbox daily



Anna Russo
ALM Consultant
[Imaginet](#)

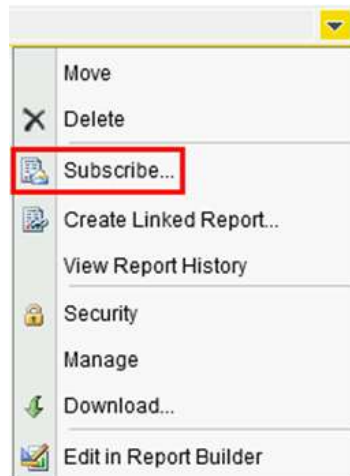
came across a TFS cube error while troubleshooting another issue that was reported to me. Now if every TFS Admin had the [TFS Admin Reports](#) emailed to them daily, then they would have seen the cube status error and responded accordingly.

It's really easy to setup Reporting Services to automatically email you the TFS Admin reports so that you are proactively managing the TFS server.

1. Navigate to Microsoft SQL Server > Configuration Tools > Reporting Services Configuration Manager
2. Fill out the Sender Address and SMTP Server

EDITOR'S NOTE! For additional information on TFS Admin Reports click on the following link for an MSDN article by Grant Holliday
<http://blogs.msdn.com/b/granth/archive/2010/07/12/administrative-report-pack-for-team-foundation-server-2010.aspx>

3. Go to the each one of the TFS Admin Reports and select Subscribe



4. Enter your email address and time of day you would like to have your report emailed.

 A screenshot of the TFS Admin Reports configuration page. It is divided into two main sections: 'Report Delivery Options' and 'Subscription Processing Options'. In the 'Report Delivery Options' section, the 'Delivered by' dropdown is set to 'E-Mail', and the 'To' field contains the name 'Anna', which is highlighted with a red box. Below this, there are fields for 'Cc:', 'Bcc:', 'Reply-To:', and 'Subject:'. The 'Subject' field contains a template string: '@ReportName was executed at @ExecutionTime'. There are checkboxes for 'Include Report' and 'Include Link', both of which are checked. The 'Render Format' dropdown is set to 'MHTML (web archive)'. The 'Priority' dropdown is set to 'Normal'. In the 'Subscription Processing Options' section, the 'Run the subscription:' section is highlighted with a red rectangle. It contains two radio buttons: 'When the scheduled report run is complete.' (which is selected) and 'On a shared schedule:'. Next to the selected radio button is a 'Select Schedule' button. Below this, the text 'At 8:00 AM every Mon, Tue, Wed, Thu, Fri of every week, starting 4/1/2013' is displayed. At the bottom, there is a section for 'Report Parameter Values' with a label 'Specify the report parameter values to use with this subscription.'

The TFS Admin Reports will be emailed to you (now if you set up an Outlook rule and ignore these emails, it kinda defeats the purpose of proactively managing the TFS server, just saying)

BTW, this also works for any of your other TFS SSRS reports that you would like automatically emailed out.

Anna Russo is an ALM Consultant and MVP with [Imaginet](#), a firm that specializes in helping companies improve software development process with TFS. She has worked with a variety of clients and upgraded/installed TFS, provided training, created custom processes, implemented release strategies, and automated build scripts. Visit Anna's blog [ImprovingSoftwareQuality.com](#)

Countering 6 typical arguments against Agile/Lean approach



Tomáš Tureček
Lean/Agile Coach
[Tieto Corporation](#)

The ambiguous nature of software development and thus the need for a different contracting approach – Agile/Lean contract. Let's now take a look at what the **typical arguments** against it are and how do we react to them.

- **Time** – “We don't have time to be involved in the development too much”
- **Fixed price** – “We need to know the cost of the solution upfront”
- **Fixed scope** – “We need to know what we are buying”
- **Delivery date** – “How can you fix a delivery date if everything keeps on changing?”
- **Risks** – “You are trying to transfer the risks to us by that open scope model”
- **Tenders**

Agile and lean contracts solve one of the root causes of tension between customer and supplier as noted here in [removing tension between customer and supplier](#).

1. Time

- We don't have time to be involved in the development so much”
- “Our people are very busy; they cannot attend demos and spend time with you on phone”

Well, what to say. Surely customers do not have time when they spend months in writing down very detailed requirements specifications at the beginning and then spend weeks or months on acceptance testing at the end to be sure that they got actually what they requested for. And after they find a result, they spend few more months on rework, arguing and in error corrections. Whether you like it or not, this is the most common work flow of traditional projects; blue parts in the picture below emphasize work of customer people during a project.

- Tender approach might be enforced by law, especially in public sector

All my articles about Agile/Lean contracts are mostly about one simple thing – we **should not commit to something we do not know** because then everyone loses if the situation gets more complicated than we expect (which usually does as there are always some **unknown unknowns**, [black swans](#)). And before we actually code some proof of concepts and tackle business and technical obstacles, we must admit that we have only limited clue about the future. This is why we should not promise something that we do not know we can deliver. Trust is hard to build but very easy to lose. So how to reduce negative impact of a tender approach?

- Can we convince the customer to use a different supplier choosing method than the tender approach? I mean, if it is not enforced-by-law?
- Can we do something in order to not commit blindly in this tender? Some proof of concept, prototype for the tender already? Can we get in touch with customer before the decision date and involve them into prototype creation? To learn more before it is too late? Working prototype might be actually a nice advantage against tender competitors. And proof of supplier competences in the area?
- Can we do a semi-commitment? To fulfill tender requirements, but not to fix everything? Like agreeing on fulfilling the business objectives but at the same time agreeing on needed support from Customer, their involvement and future easy change management?
- Tenders are not won by price only. There are many indicators that contribute to the decision about which supplier should be chosen. We can use our Agile/Lean approach as an advantage against competitors in the tender.
- The tender approach must be applied only when a certain threshold is reached. Can we agree with the customer about breaking down the functionality into smaller and cheaper chunks to avoid it?

Recently I had a discussion with a couple of customer people from one certain public sector, and they were really **sad about the law and public tenders**. They have great experience with the Agile way of working, and the current supplier (after encountering many years of failure with the traditional approach) – but the **public tender approach goes against it**. They try to avoid tendering by prolonging the current agreement they already have with supplier they are happy with.

I understand a tender is here to save costs and prevent illegal partnerships, but on the other hand it generates huge inefficiency in areas like custom software development. Hopefully the legislative will come up with an alternate option to tender someday.

[Tomáš Tureček](#) PhD, an Lean/Agile coach in [Tieto Corporation](#), has been working in IT industry for more than 15 years in various roles from development to management. He has more than 6 years of experience from coaching and mentoring deliveries in distributed environments. Tomáš contributes to IT communities, [blogs here](#) and speaks at prestigious international conferences.



Empowering Rapid Delivery of Quality Software

The e-Magazine for:

ALM - Application Lifecycle Management

DevOps - Development & Operations

Agile - Scrum, Kanban, etc.

TFS - Team Foundation Server

[Click for more Details...](#)

ALM Mag Brings You Each Month:

- **Articles, Whitepapers, Case Studies and Videos and Audio** providing:
 - Advance notice and Previews of new versions,
 - **Reviews** of new versions and updates,
 - **How To and Step-by-Step Guides**,
 - **Tips** highlighting **Primary Tools** and
 - **3rd Party Integrated or Bridging Tools**,
 - **Processes, Books and Events**
- **Surveys** to discover more about the types of content that interest you so we can serve you better
- **Industry Study and Survey Results** to provide you more insight into what others in your community are doing
- **Interviews with top ALM Leaders** to help you discover what's coming and how you can get there
- **Tutorials and Training Courses** to increase your knowledge and leverage your career
- **Event and Conference Information** with a spotlight on Details, Agenda and Calendar so you are aware of what you want to attend to increase your career potential
- ...and finally **IT Humor** (yes it does exist) including **Cartoons**

and more...

This Magazine For You If You Are:

- a CEO, CIO, COO
- an Application or Development or QA or DBA Manager or Architect
- a Development or QA Lead
- a Product Owner
- a Project Manager
- a Scrum Master
- a Business Analyst
- a Tester
- a Developer
- ...Or anyone involved in software Planning, Development, Testing, Infrastructure Provisioning or Deployment

[Subscribe](#)

[Index Page](#)

The History and Back Story...



Mike Vizados
 Certified Scrum Trainer
 Agile Consultant & Coach
implementingscrum.com

My name is [Michael Vizados](http://www.implementingscrum.com) and you may not have heard of me (it's a big world). Back in 2004 when I received my Certified ScrumMaster Certificate from Ken Schwaber, he told a story in his class about a chicken and a pig. As many people during that time, I ignored the story and it slipped my mind for a long time.

Scrum and chickens and pigs, "Yeah right," I thought. What does that have to do with *anything* about Scrum and Agile?



Zoom ahead almost ten years later and I look back at what a powerful impact this little story actually would have in my life. Like most people, this story will pass you by like the rest of the noise in our great big universe. Unlike many people, take a breath here. Pause. And think about some time in your life that things took a major *shift*.

Let's get the story out of the way first (I actually screw this up most of the times I actually *tell* this story, but through the magic of writing I should be able to get it right (even if the editors want to hack away at the actual punch line!).

A Chicken and Pig were walking down the street one day and talked about opening a morning place for breakfast. Being really entrepreneurial and pretty open to trying new ideas (think about the Lean Concept ideas heading into 2013) they decided it would be a great idea. The Chicken had a great idea for a name and of course was anxious to try out the idea on his friend The Pig. "Ham 'n Eggs". The Pig looked at him with an astonished look and told the Chicken, "No Thanks, I'd be committed you'd only be involved!"

Wow. Right?

as wrong as that *perceived* fact is!). Some people actually have this visceral reaction to the story of the Chickens and Pigs. Some people hear it and laugh it off. Some people are just learning about it (still). The story is not just associated with Scrum (much like Agile does not equal Scrum); however, it is pretty well known by now. And the story will continue to throughout the years. It's fun. It causes reactions in people. Some people laugh. Some people cry (well, actually now that I think of it I have never seen anyone actually cry over this cartoon).

Here is the important thing: The story (and cartoons) about Chickens and Pigs get people talking. Over the years the tagline for the site has evolved into, "Starting Tough Conversations about Software Development."

And that is something that is important for us to examine in this magazine — and beyond.

Within these pages (or on your screen most likely) you'll learn some new things. Maybe some of them are not so new. Here is my challenge to you (especially if you want to gloss over something because it *is* familiar to you — keep an open and beginners mind and let's see just what is possible. Take something from this magazine and apply it *today*).

Also, feel free to join me at www.implementingscrum.com if this is a topic that interests you. There are over a hundred comic strips out there and over six years of blog postings about various topics (mostly related to Scrum although sometimes I get off topic — life!). Heading into 2013 people there will join me on a new journey. Interested in joining me on this journey? The content is free.

What I am finding though is that sometimes *free* is the most expensive thing you will ever do.

Thanks for taking the time to read this overview and "my story" of where I have been. Let us know what you'd like to see examined more within the pages of this magazine that will help you move a little forward each day.

[Michael Vizados](#) is a world traveler, author, Scrum & Agile Consultant, speaker, and the person behind implementingscrum.com.

Welcome



*Keith Denham
Publisher*

*Treasure Cove Publishing
Volume 1 – Issue 2
February 2013*

ADVISORY BOARD:

*Sam Guckenheimer
Group Product Planner for
Visual Studio & Author
- Microsoft*

*Keith Pleas
Organizer & Conference
Chair - ALM Summit*

EDITORIAL REVIEW:

*Anthony Borton
ALM Ranger
ALM Trainer & Consultant
- Enhance ALM Pty, Ltd.*

*Mike Vizdos
Agile Coach
Certified Scrum Trainer
Clementino de Mendonca
National ALM Architect
- Neudesic*

Subscriptions:

almmag.com/subscribe

Email Address Changes:

support@almmag.com

Mission Statement:

*"Our mission is to promote,
support and empower
rapid delivery of quality
software."*

We're back from a very successful and enjoyable ALM Summit held in the Microsoft Conference Center on the campus in Redmond, WA. Five days and nights of terrific presentations, workshops, social events and networking with many of the brightest minds in our space.

We will be looking to bring you some commentary on the Summit in the next issue.

This February issue of ALM Mag has a definite DevOps flavor with articles from Andrew Stevens of XebiaLabs, Dr. Jacob Ukelson of Nolio and Eric Minick of UrbanCode all presenting articles on deployment strategies and processes.

The March issue is shaping up to have an even stronger Testing focus but in true Agile spirit we'll see how that plays out.

Several alliances were formed during the Summit that, while allowing ALM Mag to retain its independent position, will bring a continuity of more of the great content that we trust will be of value to your and your organizations in the months and years ahead.

For our international readers know that following the success at the Summit we have already begun to talk to conference organizers in Europe, the South Pacific and Asia about sponsoring their events in an ongoing effort to bring ALM Mag closer to you.

Our membership is growing rapidly and already in these early stages and while strong in the US, it has taken on a marked international reach.

For those of you reading this issue and who are not yet members it is not too late to take advantage of our early bird Founding Member Offer... click this link for more details.

<http://www.member.almmag.com/goto/foundingmember>

Enjoy!

Warmest Regards,
Keith Denham

Keith Denham

*P.S. Don't forget, this is an interactive magazine, tap on the audios and videos and tap on the highlighted links to see additional related content.
Tap on the Index Page Button at the foot of any page to return to the Index.*

The ALM Architect – balancing freedom with control



Dave West
Chief Product Officer
[Tasktop](#)

In his industry shaping comment, Marc Andreessen declared that [software is eating](http://online.wsj.com/article/SB10001424053111903480904576512250915629460.html) the world. <http://online.wsj.com/article/SB10001424053111903480904576512250915629460.html> Never before have so many industries, markets and services depended on software to be competitive, efficient or even useful. Software, in traditional and high-tech organizations alike, has gone from an ancillary service to a key business process.

But for many organizations, the support for this ‘key’ business process is fragmented with process, practice and tool decisions being made by development teams, departments or even individual developers. [Developer tech populism](http://blogs.forrester.com/application_development/2010/01/forrester-databyte-developer-scm-tool-adoption-and-use.html) is a growing trend where software developers (http://blogs.forrester.com/application_development/2010/01/forrester-databyte-developer-scm-tool-adoption-and-use.html) are increasingly bringing in their own tools and making decisions on process and even platform. Empowered developers and teams is a key tenant of Agile methods, allowing teams to respond to the situation, selecting the best tools, process and practices for the situation¹. But as software increases in importance to people and departments outside of IT, its status becomes worthy of management attention and has to adapt to integrate with a broader set of business processes. And there lies the rub: to be effective, software teams need freedom, but businesses who are now relying on software for competitive advantage need control, visibility and integration.

ALM is the foundation

Application Lifecycle Management (ALM) is the application of business discipline to the practice of software engineering. It is enabled by the integration of the software development tools, processes and practices in the context of management reporting. At its heart, ALM is about automating the process of software engineering, enabling everyone on a software delivery team to work effectively together. For many organizations, ALM is a dream rather than a reality, an aspiration to their tool purchasing decisions. At best, organizations have connected their Source Code Management (SCM) with their Change Management (CM), allowing change information to provide context to file changes.

Requirements, test, design and planning are still separate artifacts with limited automated integration. Management still relies on spreadsheets

¹ The Agile manifesto can be found at <http://agilemanifesto.org/>

The ALM architect needs to review each area, looking to standardize a subset of tools and define standards such reports, methods of estimation, defined schemas, and the data interchange and workflow supporting the overall process. The ALM architect should concentrate on:-

- **Management disciplines** – Look to the overall processes being used by the organization to gain understanding of the reporting and management needs. By following the same route as other business processes, such as sales and procurement, many organizations attempt to create a standardized data warehouse for software delivery. This warehouse supports the capturing of historic data over time, allowing analytics to be performed.
- **Integration** – Automating the connection between engineering and management disciplines and between the different tools within each is important for streamlining the business process. Traditional manual processes that connect disciplines, such as a spreadsheet between testing and development or documents being emailed from requirements to development, do not scale or enable management reporting to be created. ALM tool platforms, such as IBM Jazz and MS Visual Studio, go some of the way to provide this interchange platform, but for the majority of organizations with heterogeneous tools, it will require looking into lifecycle integration technology.
- **Engineering discipline** – It would be true to say that not all disciplines are created equal, that certain engineering disciplines are more ready for an ALM approach. In particular, focus on the biggest areas of pain or areas where change is already happening because of other initiatives, such as continuous delivery or Agile development.

Conclusion

If your organization thinks that software delivery and maintenance is a key business process, it is crucial that they invest in the role of the ALM architect. That role provides ownership of the practical implementation and long term view of ALM within the organization. The architect should be measured by the success of ALM in delivering both management discipline to the practice of software delivery and maintenance and by realizing faster time to market or reduced cost.

By applying classical architectural skills, the ALM architect can build a roadmap and implementation plan that provides short-term benefits whilst moving the organization onto a more formal, integrated, transparent delivery and maintenance organization. This would allow executive management to make decisions in real time and plan and respond to market opportunities faster. For many organizations, software delivery slows down business change. By improving the business process of software, organizations will be better equipped to deliver and compete in the 21st century, which for many is the age of software...

Dave West is the Chief Product Officer at [Tasktop](#), where he engages with customers and partners to drive Tasktop's product roadmap and market positioning. Recently David served four years as vice president, research director at Forrester Research. Previously David led the development of the Rational Unified Process (RUP) for IBM/Rational. David authored the acclaimed book: [Head First Object-Oriented Analysis and Design](#). On Twitter follow Tasktop [@Tasktop](#) or follow Dave West [@DavidJWest](#)

Professional Scrum Development with Visual Studio 2012



Richard Hundhausen
President
[Accentient](#)

Scrum and Visual Studio 2012 make for a powerful combination. Organizations that can leverage both will see an increase in the quality of their products as well as the velocity of their teams. In this webcast we will explore the Scrum framework, the Visual Studio Scrum process template, and the planning and tracking tools in Team Foundation Server 2012.

Recommended Audiences

Business Decision Makers, Developers, Technical Decision Makers

Streaming Video only available when connected to the Internet

When Online tap this screen icon to play the video
Once playing, tap the full screen icon and rotate your tablet



[Richard Hundhausen](#) is the president of [Accentient](#), a company that helps software development teams understand and leverage Microsoft ALM tools and Scrum. He has over 30 years of software development experience and over 20 years of training experience. Richard is a Microsoft Regional Director, a Visual Studio ALM MVP, and author of *Professional Scrum Development with Microsoft Visual Studio 2012* by Microsoft Press.

Why Small Companies Need ALM More



Andrew Clear
ALM Consultant
[Northwest Cadence](#)

W

e hear it a lot. "Oh, we're too small to need all that." But the truth of the matter is that, in all likelihood, you do need that. You see, modern ALM best practices are all about one thing: keeping your project from failing.

Let me run a couple scenarios by you.

You're a project manager at a Fortune 500 company worth a couple hundred million dollars. You're about to budget \$200,000 on a six month software project and you're going to use a typical waterfall methodology. When that project falls flat on its face and gets scrapped without ever delivering anything of actual value you'll hem and haw, point out that this or that made the project unfeasible to begin with and point out that the other three projects you're managing are going great. No one is really happy about the project failing at the moment, but a few months from now no one will remember it anyways.

Now, consider you're a member of a small startup. You only have five developers (in fact the startup only has six employees), and you're about to budget that same \$200,000 on a six month software project. Of course, your company doesn't actually have a net value at the moment, and the money you're spending is directly from one of your primary investors who also has an interest in the project itself. What happens if this project fails without delivering anything of value? The investor backs out, the company goes belly up, and you all get to play the employment game.

These scenarios, while obviously exaggerated, are not all that far fetched. A project failing without delivering any value can be *devastating* to a small to mid-sized company. The [statistics tell a grisly tale](#): most software projects fail. Most of those that fail do so because of communication and requirements gathering issues. Most of the budget is spent fixing "self inflicted" problems, and most bugs are found by the end users. Stakeholder feedback comes too late to do anything about it.

These are *exactly the issues that ALM addresses*. Work item tracking increases communication and visibility. A prioritized backlog allows stakeholders to provide feedback *before* requirements are implemented. Unit testing, automated coded UI testing, repeatable test cases, and continuous integration help find bugs before they get into production. A tool like Visual Studio's Feedback Client allow stakeholder feedback during development.

Perhaps most importantly, the “best practices” associated with modern ALM ensure that we aren’t *creating legacy code*. Maintenance is a huge cost that is rarely adequately planned for. Taking short cuts, and writing code that “just works” almost ensures that whatever you budgeted for maintenance won’t be enough, because skipping things like unit tests and code reviews practically ensures that you are actively creating legacy code.

But all of those little bits are easy to skip when you’re part of a small company. Creating that initial product is faster, and in most cases “easier” when you skip them. Landing that venture capital cash is all important, so you create content as quickly as possible. You think “we need to land this VC quick – then *after* we get the funding we’ll be able to rewrite it properly and fix all those problems”. In that situation you’re almost *worse* off if you *do* get the funding. No VC is going to give you \$1M to rewrite the same thing you just showed them. You can’t go to your investors and admit that the thing you just sold them doesn’t actually work. They expect that you’re going to *build upon* what they’ve seen, not *rebuild* it.

Let me return to a point I glossed over previously. Software projects that fail *without delivering any value* are devastating to small companies. The resources just aren’t there to scrap an entire project and start over. But modern ALM espouses *continuous value delivery*; implementing requirements incrementally, the completion of which creates a unit of value. This means you can start seeing returns on your project before it is

“complete.” Then if (when) you need to pivot, you don’t have to start back at square one, and all that work wasn’t a complete waste.

One final aspect of ALM utopia to discuss. Tech startups quite often become victims of their own success. Someone has a great idea, a team of development ninjas throw it together as quickly as possible. Cash starts flowing in, as do the customers. But then the customers start clamoring for new functionality. And they want it now. So your dev team grows, throwing legacy code on top of legacy code to keep that feature deployment rate high. The original ninjas start whispering that the infrastructure can’t support the weight, but the process is more or less still working. Right up until it stops working. And in another month you end up another [MySpace](#) because you are unable to pivot. Unable to control the exponential growth of technical debt.

The easiest time to implement the complete set of ALM best practices is *before* you start your project. Or as close to the beginning as possible. The longer you’ve been piling legacy code on top of legacy code the harder it is to implement process changes. So the next time you hear yourself saying, “we’re too small for that” take a step back and reconsider. No matter where you are in your development life, *there won’t ever be a “better” time to modernize your practices and processes*. It won’t get any easier, it will only get harder. Of course, wait too long and it won’t be an issue at all.

Andrew Clear, ALM Consultant at [Northwest Cadence](#), is dedicated to bringing agile to the agile-less.



[Northwest Cadence, Inc.](#) provides training and consulting services across the United States. The exclusive focus on Software Process and Microsoft ALM tools ensures clients benefit from deep technical expertise, sound process acumen, and hundreds of practical experiences on ALM projects. They partner with software development teams to improve processes and business results.

Enacting Scrum and Agile with Visual Studio 2012



Clementino de Mendonca
National ALM Architect
[Neudesic](#)

In any conversation about Agile, it's always a good idea to set the ground by first revisiting the Agile Manifesto:

We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

- *Individuals and interactions over processes and tools*
- *Working software over comprehensive documentation*
- *Customer collaboration over contract negotiation*
- *Responding to change over following a plan*

That is, while there is value in the items on the right, we value the items on the left more.



I want to emphasize the last phrase: “That is, while there is value in the items on the right, we value the items on the left more.” So there is value on the items on the right. I have seen in several conversations with my Agile peers that they tend to sometimes forget that there is value in processes and tools, in doing some documentation, in negotiating a contract and in following a plan. It is just that when push comes to shove, if you have to make a decision, you have to value more the input that you get from individuals and interactions, from “kicking the tires” of working software, from collaborating with a customer and finding out face to face what they want exactly, and finally being flexible to respond to change when the time comes.

But how does TFS enable us to follow the Manifesto, to being Agile?

through the help of tools such as Visual Studio 2012, which is currently the best ALM tool suite available.

Tools cannot make your team work in an Agile manner. Only after the ideas contained in Agile and Scrum are understood and learned is that your team will achieve proficiency, and will be able to consciously use the tools for their intended purpose. Until then tools will help you get started through some of the built-in processes they enable, but their best use will be as canvasses to your team's creativity and ongoing improvement efforts, not as silver bullets that would somehow take your team out of the morass of waterfall.

Acknowledgement:

My thanks to William Salazar, Wayne McDonald and Gary Pedretti for conversations which influenced this article.

Clementino de Mendonça is a Professional Scrum Developer Trainer, a [Neudesic](#) National ALM Architect, a former National Lead for ALM and Testing with Capgemini/Sogeti, and was on the Microsoft Services National ALM Team for 4 years. On the Visual Studio Process Team in 2007, he helped design the Visual Studio 2010 process templates (MSF Agile 5.0 and MSF CMMI 5.0).

Podcast Podium

Interviews with Agile Evangelists



Derek Neighbors
[Integrum Technologies](#)

[Agile Weekly PodCast](#)

Episode #95 – Moving to a Team Room



Clayton Lengel-Zigich, Roy van de Water, Isaac, Deepa & Ryan discuss:

Moving from cubes to team rooms

[Derek Neighbors](#) is a serial entrepreneur who helps people bring ideas to reality. Derek co-founded [Gangplank](#), a collaborative workspace, in 2008 to encourage local “creatives” to explore innovative ideas. A partner at [Integrum Technologies](#), a consulting services firm helping companies build high performing teams, Derek teaches entrepreneurship at [Arizona State University](#).

Enterprise Application Delivery and the Cloud



Dr. Jacob Ukelson
VP Product Strategy
Nolio

I have been spending a lot of time lately looking at the cloud from an operational perspective with respect to applications – I guess it would fall under the general banner that some analysts would call DevOps and others would call AppOps. I see the difference between the two as either looking at applications from the perspective of everything that needs to be done before you can deploy an app, the other looks more at everything that needs to be done to deploy an app and monitor it afterwards. The line between them is really fuzzy – and as the cloud becomes more production oriented it will become even fuzzier.

What I found is that actual enterprise production applications in the cloud (not SaaS applications) are few and far between - so not a lot of attention has been paid to the lifecycle issues of managing enterprise production applications in the cloud. Dev and QA are the kings of cloud in the enterprise at the moment. I also found that as opposed to the [NIST definition of cloud computing](#) – IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) which seem to describe a nice progression of functionality for the cloud – the real world is much messier. SaaS came first and most SaaS providers didn't build their applications on IaaS or PaaS - they built their own homebrew "Private PaaS" tailored to their specific application using a mixture of bespoke and off-the-shelf tooling. I think that enterprise production applications will look very similar - just that they will use off-the-shelf IaaS for infrastructure provisioning and off-the-shelf PaaS for specific components in their application stack.

As I was learning all this I think I finally understood why the cloud matters – way beyond its value as a cheaper delivery model, or a way to save on infrastructure costs. Cloud will enable IT to provide application delivery as an agile production line from dev to delivery. I use the term production line, but the cloud actually holds the promise of being able to provide much more than a physical production line – application update lifecycles of days or hours, not months or years. I think as this picture becomes whole – it will drastically change the way we think about applications.

[Dr. Jacob Ukelson](#) is an expert in developing innovative solutions to real-world customer challenges. He is VP of Product Strategy at [Nolio](#) - the Zero Touch Deployment™ company. Nolio's Application Release Operations platform reduces time-to-market and makes enterprise operations cloud ready. [Click here](#) to read Nolio's Blog.

Bridging Language Gaps and Creating Executive Summaries Using VS Test Professional 2012



Tara Charter
Sr. Agile Business Analyst

feel like I'm speaking two different languages when I deal with the company itself and the developers working within the company. I also think developers become frustrated with me when they can't immediately determine the origin of a requirement.

In *Seven Steps to Mastering Business Analysis*, one developer reveals frustration with the business analysis process:

I began my career as a developer and was successful. I found that not only did I enjoy solving problems but enjoyed finding them and learning about business processes from the business experts. I remember the first time that I was programming based on requests and specifications from a BA. I felt frustrated because I wasn't tasked with talking with the users directly. I wanted to know exactly what went on in the business and why they had requested this software functionality. (Carkenord, 2009, B2T Training)

One of the most common communication challenges and often a barrier to delivering excellence on a project is the ability to speak multiple languages. As a business analyst or product owner, I must speak several languages throughout the day depending on my audience.

Each time I communicate, I must tailor my communication to my audience to avoid communicating ineffectively and wasting valuable time. And the language for each needs to be appropriate to their levels of expertise and understanding. Communicating with the developers must be crystal clear thus enabling the developers to create the system or application required by the company ([Project Community, Louise Ledbrook, November 18, 2011](#)).

Microsoft Visual Studio Team Foundation Server (TFS) 2012 provides

[Tara Charter](#), a Sr. Agile Business Analyst, greets every new business challenge as an opportunity to make a difference. Tara knows the only way to truly solve a business' unique problems is to really understand what the customer really wants.

Testing in the modern application lifecycle



Martin Hinshelwood
ALM Consultant
[Northwest Cadence](#)

Manual Testing in this new age of the modern application lifecycle has taken on new complexities that make it even more difficult to track and identify which tests are passing, which are failing and to which environment that data should be associated.

As we move into, or you might already be there, the era of agile and the delivery timetable shrinks to only a few weeks we can no longer do all of our testing in isolation. We need to work with the Engineers, Analysts and Operations on Development Teams and effectively swarm to solve the problems. Whether that problem is adding new features, fixing a bug or deploying to an environment it is now in the scope of all the departments and this new world does not recognize even the idea of the silo.

The Modern App Lifecycle

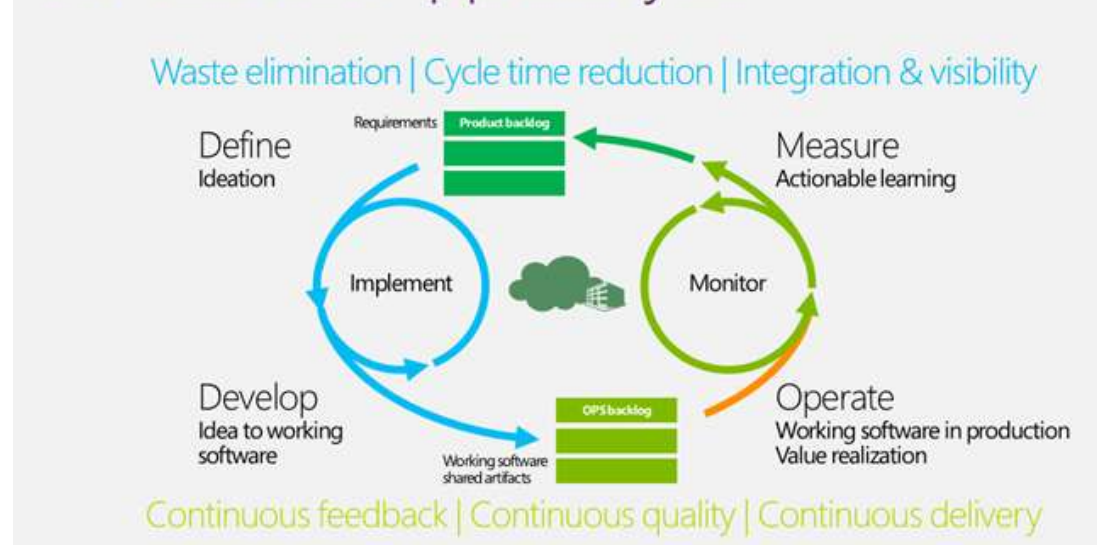


Figure: Manual Test makes sure that we build what we need and no more

It is no longer acceptable to “test quality In” as this is far too expensive

Martin Hinshelwood is an ALM Consultant at [Northwest Cadence](#), and [Visual Studio ALM MVP \(2013\)](#). He was named [ALM MVP of the year 2011](#), [ALM Ranger Champion for 2011](#) and was one of the first [Professional Scrum Trainers](#) with Scrum.org. Martin regularly delivers consulting and training while writing for his [Visual Studio ALM blog](#), and is a sought-after speaker.



www.unicom.co.uk

UPCOMING CONFERENCES AND WORKSHOPS

Application Lifecycle Management & Recent Trends in Process Improvement

28 February, Co-Located London

Special Offer: Book 1 delegate place & get another 1 free
Quote 'ALMMag' **Register Now**

DevOps Deep Dive

21 February, Amsterdam

DevOps Summit

23 May, London

**Save £100 with our
Early Bird Deals**

Agile in the Public Sector & Agile in the Finance Sector

7 March, Co-Located London



+ 44 (0) 1895 819 489

Request a Call



info@unicom.co.uk



[@UNICOMSeminars](https://twitter.com/UNICOMSeminars)



[UNICOM Seminars](https://www.linkedin.com/company/unicom-seminars)



www.facebook.com/unicom.seminars

Index Page

Continuous Delivery: Delivering IT to the Always Up-to-date User



Andrew Phillips
XebiaLabs

Trying to serve today's always connected, always updated IT user with anything less than a fully automated application delivery process makes no sense. Software Delivery is evolving from an effort-intensive, error-prone "big bang" activity to a seamless, incremental and continuous process that is proving to be faster, more reliable and more efficient.

With today's always-connected user, able to access your services all the time, everywhere, and to evaluate your competitors' offerings with a few taps or clicks, the business of service delivery is never "done". Your service is expected to be always available and continuously improving.

Businesses understand that they need to be "always-on". This requires constant, consistent application and service delivery to ensure a positive, seamless, rich customer experience.

Certain software delivery and integration strategies are central to allowing your organization's offering to be reliable and engaging, meeting and exceeding the high expectations of today's audience that demands its content and applications be constantly available and instantly gratifying.

Expanding on from the widespread adoption of Agile development practices in the enterprise, the leading strategy to achieve this today is *Continuous Delivery*.

Continuous vs. Traditional Manual Delivery

In most organizations, the software delivery process today involves many manual or semi-automated steps, requiring a significant investment in terms of scripting and troubleshooting. Further, the delivery and release process followed varies widely across departments, applications and technologies, leading to error-prone, non-transferable activities and costs that increase in direct proportion to the release volume.

Continuous Delivery is a fundamental shift in the software development and delivery process, based on a simplified release management practice that unifies development, production testing, QA, testing and operations



POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with XebiaLabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

XebiaLabs
the deployment automation company

Discover the XebiaLabs
Continuous Delivery Platform.
www.xebialabs.com

faster pace, features will land in your customers' hands much more quickly, and with a far lower failure rate. This delivers not only the latest and greatest your company has to offer, but also aligns the continuous delivery of software with today's connected culture giving rise to a seamless customer experience and greater innovation.

Making the Transition

Beginning the process of changing over to Continuous Delivery begins with an honest assessment of your existing model and your future needs. This will help you determine the best order and approach for your implementation, and give rise to realistic expectations and deadlines.

Engaging your development team is an essential element to ensuring a positive shift into automated application delivery. Not only

will their insight and experience be invaluable for understanding the challenges you have to overcome, their participation is required to accurately determine the best strategies for optimizing a new process.

The next step is to review technologies that underpin a move to Continuous Delivery, including tools for Continuous Integration, automated testing, provisioning and automated deployment. Oftentimes, organizations are finding they have some of these tools in-house already, and it is a matter of adding some final pieces to make the move to Continuous Delivery.

Once the elements are all capable of working together in order to deliver improved functionality, continuous application delivery becomes the new standard.

[Andrew Phillips](#) is Vice President, Product Management at [XebiaLabs](#), a global provider of Application Release Automation offering a Continuous Delivery platform to help enterprises improve and accelerate the application delivery process. To get started moving to Continuous Delivery, [Read the new white paper](#).

Modern Requirements Management



inteGREAT & TFS share a single database

Key Features that extend TFS capabilities

Adaptive project methodology support	Completely configurable taxonomy	Project baselining	Automatic document generation
Automatic test case generation	Reusability across projects	Impact assessment	Author and review in WORD
Simulations and on demand Wireframes	Non-Functional requirement library	Automated Function Point estimation	Specification Checker to validate requirements
Wizard authoring for completeness	Legacy Word & Excel import	Conceptual data model creation	Drag & drop traceability

Countering 6 typical arguments against Agile/Lean approach



Tomáš Tureček
Lean/Agile Coach
[Tieto Corporation](#)

The ambiguous nature of software development and thus the need for a different contracting approach – Agile/Lean contract. Let's now take a look at what the **typical arguments** against it are and how do we react to them.

- **Time** – “We don't have time to be involved in the development too much”
- **Fixed price** – “We need to know the cost of the solution upfront”
- **Fixed scope** – “We need to know what we are buying”
- **Delivery date** – “How can you fix a delivery date if everything keeps on changing?”
- **Risks** – “You are trying to transfer the risks to us by that open scope model”
- **Tenders**

Agile and lean contracts solve one of the root causes of tension between customer and supplier as noted here in [removing tension between customer and supplier](#).

1. Time

- We don't have time to be involved in the development so much”
- “Our people are very busy; they cannot attend demos and spend time with you on phone”

Well, what to say. Surely customers do not have time when they spend months in writing down very detailed requirements specifications at the beginning and then spend weeks or months on acceptance testing at the end to be sure that they got actually what they requested for. And after they find a result, they spend few more months on rework, arguing and in error corrections. Whether you like it or not, this is the most common work flow of traditional projects; blue parts in the picture below emphasize work of customer people during a project.

- Tender approach might be enforced by law, especially in public sector

All my articles about Agile/Lean contracts are mostly about one simple thing – we **should not commit to something we do not know** because then everyone loses if the situation gets more complicated than we expect (which usually does as there are always some **unknown unknowns**, [black swans](#)). And before we actually code some proof of concepts and tackle business and technical obstacles, we must admit that we have only limited clue about the future. This is why we should not promise something that we do not know we can deliver. Trust is hard to build but very easy to lose. So how to reduce negative impact of a tender approach?

- Can we convince the customer to use a different supplier choosing method than the tender approach? I mean, if it is not enforced-by-law?
- Can we do something in order to not commit blindly in this tender? Some proof of concept, prototype for the tender already? Can we get in touch with customer before the decision date and involve them into prototype creation? To learn more before it is too late? Working prototype might be actually a nice advantage against tender competitors. And proof of supplier competences in the area?
- Can we do a semi-commitment? To fulfill tender requirements, but not to fix everything? Like agreeing on fulfilling the business objectives but at the same time agreeing on needed support from Customer, their involvement and future easy change management?
- Tenders are not won by price only. There are many indicators that contribute to the decision about which supplier should be chosen. We can use our Agile/Lean approach as an advantage against competitors in the tender.
- The tender approach must be applied only when a certain threshold is reached. Can we agree with the customer about breaking down the functionality into smaller and cheaper chunks to avoid it?

Recently I had a discussion with a couple of customer people from one certain public sector, and they were really **sad about the law and public tenders**. They have great experience with the Agile way of working, and the current supplier (after encountering many years of failure with the traditional approach) – but the **public tender approach goes against it**. They try to avoid tendering by prolonging the current agreement they already have with supplier they are happy with.

I understand a tender is here to save costs and prevent illegal partnerships, but on the other hand it generates huge inefficiency in areas like custom software development. Hopefully the legislative will come up with an alternate option to tender someday.

[Tomáš Tureček](#) PhD, an Lean/Agile coach in [Tieto Corporation](#), has been working in IT industry for more than 15 years in various roles from development to management. He has more than 6 years of experience from coaching and mentoring deliveries in distributed environments. Tomáš contributes to IT communities, [blogs here](#) and speaks at prestigious international conferences.

Deployment Automation Basics

Implementing Consistent Deployments Across Environments



Eric Minick
[UrbanCode](#)

T organizations are finding it difficult to release more quickly without adding more people and incurring additional risk. Here we discuss the limitations of manual deployments, common failure patterns, the benefits of automation, and a base set of features an automated system should provide.

Key Findings

Manual deployments are inherently slow and error prone. Deployment automation used only in development or only in operations may help one silo, but leads to a hand-off where changes to the process may be insufficiently communicated.

Automated deployments provide superior audit trails.

An automated deployment infrastructure provides a framework to build upon. Additional activities such as automated functional testing can leverage the deployment infrastructure.

Deployments standardized across environments must still take into account environmental differences. The environment configuration is a key concern.

Introduction

In a continuing poll, over 75% of respondents identified their deployment process as either "Entirely Manual," "Mostly Manual," or "Mostly Scripted." Relatively few indicated that they had push-button deployment capabilities. This result is not surprising: Until recently, deployment automation options have been sparse, with vendors offering "solutions" that do not scale to the enterprise; require considerable process reengineering to implement; and provide little visibility into the "who, what, when, where, why and how" of deployments.

Companies have turned to internal solutions, and more scripting, to help their team members achieve more consistent deployments and keep pace with the increasing demands of competitive markets. However, as with many "home grown" solutions, the expense ratio, measured in both time and dollars, often proves high: Most organizations are not in the business of providing "process automation," and have to borrow

Eric Minick is a lead consultant at [UrbanCode](#) where he helps customers get the most out of their build, deploy and release processes. Eric has been at the forefront of Continuous Delivery for 8+ years in roles ranging from developer to tester to release consultant.

It's Time for Test Management in the Cloud



Mush Honda
QA Director
[KMS Technology](#)

When it comes to testing, what's up with the cloud?

Cloud services have really captured the imagination of the business world. According to a Gartner survey 71% of organizations have adopted SaaS (Software as a Service) within the last three years.

Quality Assurance teams however, seem to have missed the boat, barely leveraging the benefits of the cloud. While enterprises and their so-called "QA Verified" software may be cloud-based, Test Management itself generally isn't.

Advantages of Test Management in the cloud

The benefits a cloud-based approach brings are plentiful and are particularly pertinent for QA teams. Top of the list for most companies adopting SaaS is the cost-savings. The ability to eradicate setup and ongoing maintenance costs for your QA toolset and data storage is most welcome. Working via a browser means no special software/hardware overhead.

Consider the nature of most test projects and you'll find that they involve disparate groups working in remote locations, often around the clock. A central repository that remains accessible for everyone with solid SLAs (service level agreements) ensuring zero or minimal downtime is a vast improvement over the traditional approach. This increase in reliability comes alongside solid security encryption for peace of mind.

A snapshot of cloud-application benefits include:

- No on-going infrastructure management, operational or installation costs incurred. Backup and recovery is relatively simpler and cheaper than traditional methods, as service providers offer recovery services;

Mush Honda is QA Director at [KMS Technology](#), a provider of IT services across the software development lifecycle with offices in Atlanta, GA and Ho Chi Minh City, Vietnam. He has previously played various QA roles at Ernst & Young, Nexidia, Colibrium Partners and Connecture. KMS services include application management, testing, support, professional services and staff augmentation.



Empowering Rapid Delivery of Quality Software

The e-Magazine for:

ALM - Application Lifecycle Management

DevOps - Development & Operations

Agile - Scrum, Kanban, etc.

TFS - Team Foundation Server

[Click for more Details...](#)

ALM Mag Brings You Each Month:

- **Articles, Whitepapers, Case Studies and Videos and Audio** providing:
 - Advance notice and Previews of new versions,
 - **Reviews** of new versions and updates,
 - **How To** and **Step-by-Step Guides**,
 - **Tips** highlighting **Primary Tools** and
 - **3rd Party Integrated or Bridging Tools**,
 - **Processes, Books and Events**
- **Surveys** to discover more about the types of content that interest you so we can serve you better
- **Industry Study and Survey Results** to provide you more insight into what others in your community are doing
- **Interviews with top ALM Leaders** to help you discover what's coming and how you can get there
- **Tutorials and Training Courses** to increase your knowledge and leverage your career
- **Event and Conference Information** with a spotlight on Details, Agenda and Calendar so you are aware of what you want to attend to increase your career potential
- ...and finally **IT Humor** (yes it does exist) including **Cartoons**

and more...

This Magazine Is For You If You Are:

- a CEO, CIO, COO
- an Application or Development or QA or DBA Manager or Architect
- a Development or QA Lead
- a Product Owner
- a Project Manager
- a Scrum Master
- a Business Analyst
- a Tester
- a Developer
- ...Or anyone involved in software Planning, Development, Testing, Infrastructure Provisioning or Deployment

[Subscribe](#)

[Index Page](#)

Scrum – It's STILL not a Silver Bullet!



Mike Vizdos
 Certified Scrum Trainer
 Agile Consultant & Coach
implementingscrum.com

S

crum is not a Silver Bullet that will fix everything in your organization. I've been saying — and writing — about this for many years (look at the copyright of this cartoon!). Take a look at some of the themes in this magazine (DevOps and more), along with the different tools and methodologies that offer possible promises to fix what hurts today (including TFS, Scrum, and others).

My suggestion is to continue reading, reviewing and digesting this information.

It is good stuff; however, don't think for a moment that anything will be a Silver Bullet in your organization.

There is no such thing as a Silver Bullet. Remember this.

If something sounds like it is too good to be true, it probably is. We all know these statements make logical sense... but in reality we do not always deal with logic. If you are looking for a quick fix, one does not exist. Change is hard. Deal with it. In reality.



By Clark & Vizdos

© 2006 implementingscrum.com

For Reprints of this page
 And previous strips

[Click Here](#)

[Michael Vizdos](#) is a world traveler, author, Scrum & Agile Consultant, speaker, and the person behind implementingscrum.com.

Welcome



*Keith Denham
Publisher*

*Treasure Cove Publishing
Volume 1 – Issue 1
January 2013*

Welcome to the second issue of ALM Mag, the only online e-magazine dedicated to focus on topics of interest to the ALM, DevOps, Agile, Lean, QA, Testing, Configuration Management, Deployment and TFS community.

This column will be an occasional addition to an edition.

ALM Mag the magazine will be available on desktops but primarily aimed at the three major mobile tablet platforms: pcPad, aPad and iPad.

I wish you many hours of enjoyment over the coming issues as you build a library of knowledge and resources you can turn to support you daily.

Warmest Regards,
Keith Denham

ADVISORY BOARD:

*Sam Guckenheimer
Group Product Planner for
Visual Studio & Author
- Microsoft*

*Keith Pleas
Organizer & Conference
Chair - ALM Summit*

EDITORIAL REVIEW:

*Mike Vizdos
Agile Coach
Certified Scrum Trainer*

*Clementino de Mendonca
National ALM Architect
- Neudesic*

Subscriptions:

almmag.com/subscribe

Email Address Changes:

support@almmag.com

Mission Statement:

*"Our mission is to promote,
support and empower
rapid delivery of quality
software."*

P.S. Don't forget, this is an interactive magazine, tap on the audios and videos and tap on the highlighted links to see additional related content.

Tap on the Index Page Button at the foot of any page to return to the Index.

Keith Denham

The Principles of Agile Architecture



**Alex Yakyma and
Dean Leffingwell**
[Scaled Agile Framework](#)



*with contributions from
Ryan Martens and
Mauricio Zamora*



You've got to think about big things while you're doing small things, so that all the small things go in the right direction.” – *Alvin Toffler*

Abstract

Large enterprises undertake large software initiatives to be able to compete in the market. Indeed, they become increasingly dependent on software for their very business success. However the increasing complexity of the solutions they build, and their constant aging in the face of rapid technologic and business change, make the systems, and the enterprise that are beholden to them, less responsive over time. The systems become brittle, hard to maintain, and hard to evolve to meet the changing needs of the enterprise and the external marketplace. This jeopardizes future success, as the software legacy systems can no longer be quickly evolved to innovate—nor even minimally meet—the needs of the next generation of business opportunities.

The Scaled Agile Framework is designed to help enterprises address the current and future software solution challenges by providing a set of proven patterns enterprises can use to build new software assets and more rapidly evolve current systems. Core to SAFe is an understanding and emphasis on the critical importance that *software architecture* plays in helping to assure business success. In this article we reprise and revise a set of *seven principles of agile architecture* designed to help the enterprise reason about software design and software architecture, and balance the role that these two different elements play in building and revising enterprise class software systems.

[Alex Yakyma](#) is an Associate Scaled Agile Framework methodologist and agile consultant, trainer and who has been applying and advancing agile practices for over a decade. Alex fluent in English, Russian, Java, C++, TDD and ATDD, works in developing and applying the framework for multinational clients in the US, Europe and Asia.

[Dean Leffingwell](#), software industry veteran and Lean Systems Society Fellow, is a renowned methodologist, author, coach, entrepreneur and executive, having founded numerous high tech companies. He most recent books are [Agile Software Requirements](#) and [Scaling Software Agility](#). His current project is the [ScaledAgileFramework.com](#), which describes a comprehensive system for scaling Lean and Agile practices to the largest software enterprises.

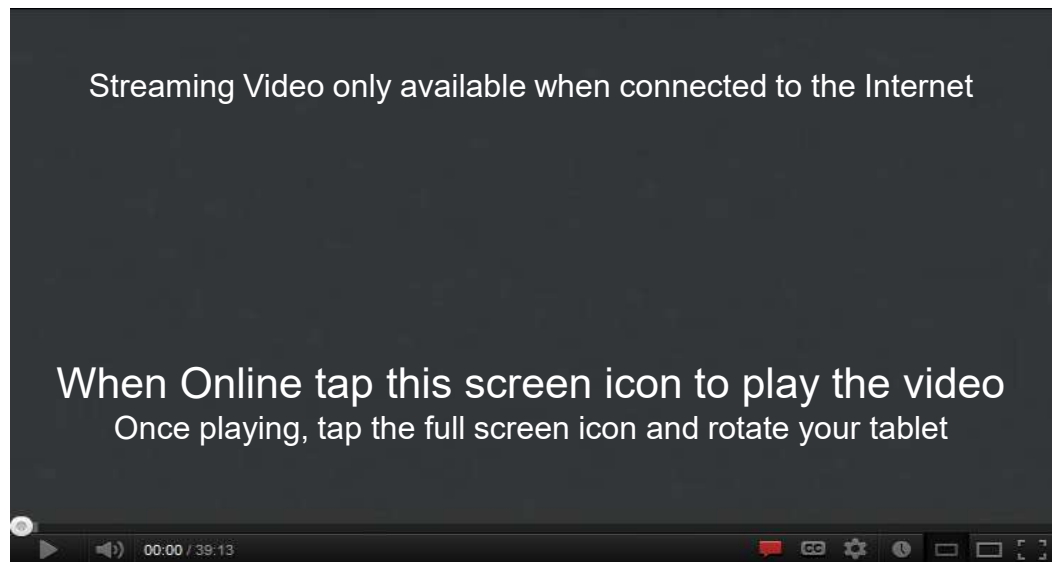
Deliver Early, Often, and Exceptionally with Visual Studio Premium



Cheryl Hammond
ALM Consultant
[Northwest Cadence](#)

Developers

Expect to see: new features f



Click on the graphic above to play the video
If the graphic above doesn't play for you
[click here for a dedicated player and the option to Download](#)



[Cheryl Hammond](#) leads the [Northwest Cadence](#) Practices Team, focusing on the ways in which agility, architecture and test practices improve the lives of developers, strengthen teams, and deliver better software. Cheryl has authored Microsoft courseware and white papers on end-to-end ALM with Visual Studio 2012, and recorded videos on modern application development including Windows 8 applications and C++. Cheryl believes that anything is possible.

[Northwest Cadence, Inc.](#) provides training and consulting services across the United States. The exclusive focus on Software Process and Microsoft ALM tools ensures clients benefit from deep technical expertise, sound process acumen, and hundreds of practical experiences on ALM projects. They partner with software development teams to improve processes and business results.

Brian Harry Discusses ALM and TFS



Brian Harry
Technical Fellow
[Microsoft](#)

We're here today talking with Brian Harry Technical Fellow at Microsoft and a person who's been really guiding and leading the effort to develop TFS and the ALM platform into a really all-encompassing and totally integrated platform for software development right from the very beginning.

For the full unedited interview click the following audio player graphic.



Click on the graphic above to play the interview

If the graphic above doesn't play for you
[click here for a dedicated player and the option to Download](#)

ALM Mag: Brian thanks for sitting with ALM Mag today and sharing your time and experience. Firstly, I know that you have a farm. How do you manage to balance family life with your work life here at Microsoft and still have a working farm as well?

Brian Harry: *It's a question my wife asks me as well. The farm is my release. Some people watch TV, or movies, or biking, or boating, or whatever their passion is. My downtime passion is farming. I like it: I like being outdoors, I like taking care of the animals, I even like digging the ditches, it's cathartic for me. That's how I relax.*

ALM Mag: Is this a commercial farm?

Brian Harry: *Yes it's a commercial farm. My wife runs the business. We've got about a hundred head of cattle and so it's mostly beef that we sell. We also have an orchard and sell fruit. We sell a couple of thousand pounds of fruit a year, so it's fun. We have a good time with it. It keeps us really busy.*

ALM Mag: Probably most people don't know about your origins and how you evolved to the Microsoft path was originally through da Vinci systems with your work on email platforms.

[Brian Harry](#) is a Microsoft Technical Fellow working as the Product Unit Manager for Team Foundation Server for which he is a key driver and visionary. Brian's career started with work on email platforms for Da Vinci Systems, then he co-founded One Tree Software a startup that created SourceSafe and brought it to Microsoft in 1994 where it became Visual SourceSafe.

Microsoft ALM Certification

– Part 1 of 3



Anthony Borton
ALM Trainer & Consultant
[Enhance ALM](#)

Welcome to the first of a three part series I'm writing to help readers learn about Microsoft's new ALM certification and the three exams required to gain this certification.

They are: 70-496, 70-497 and 70-498

In this article I'll introduce you to the new Microsoft Certified Solution Developer: Application Lifecycle Management certification and then focus on the 70-496 exam. The second article will focus on exam taking tips and then look closely at exam 70-497. The final article will cover exam 70-498 and how to make the most of your new certification.

Microsoft Certified Solution Developer: Application Lifecycle Management



Application Lifecycle Management

At the ALM Summit in Redmond in February 2013, Microsoft publically announced the Microsoft Certified Solution Developer: Application Lifecycle Management certification. While the certification has been available for a couple of months, this was the first public announcement. It was also confirmation of the re-birth of the MCSD certification which had previously been retired in favour of the Microsoft Certified Professional Developer (MCPD) in the 2010 timeframe.

The MCSD: Application Lifecycle Management certification requires the successful completion of the following three exams.

- 70-496: Administering Microsoft Visual Studio Team Foundation Server 2012
- 70-497: Software Testing with Visual Studio 2012
- 70-498: Delivering Continuous Value with Visual Studio 2012 Application Lifecycle Management

[Anthony Borton](#) is an ALM trainer and consultant for [Enhance ALM](#), an Australian based company specializing in helping organizations successfully adopt ALM best practices. He is the author of eight TFS 2012 training courses and delivers the most comprehensive range of TFS courses in the world today. You can find out more about these courses in the United States at www.quicklearn.com.

TF.exe Fast Overview

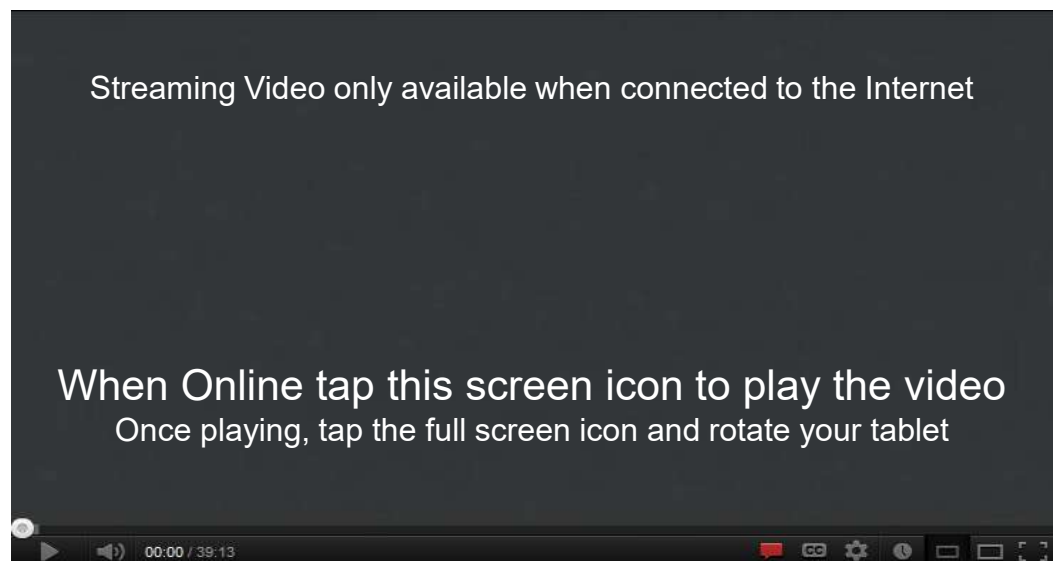


Mohamed Radwan
Senior ALM Consultant
[Marvel ALM](#)

will give a fast overview about tf.exe command line, I'll show how to use different options, like check-in, workspaces, view, diff, status and workfold.

The most important consideration that we have take is to navigate to the mapping folder inside the working machine so we can use different options otherwise we will see the following: error

The following is a step-by-step video for the post



Click on the graphic above to play the video

If the graphic above doesn't play for you

[click here for a dedicated player and the option to Download](#)

[Mohamed Radwan](#) is a Visual Studio ALM MVP, Senior ALM Consultant at [Marvel ALM](#) with more than 10 years of software industry experience, he focuses on C#, MVC, JQuery, BDD, TDD, TFS and Practical Agile Implementation. A founder of [TFSEG](#) User Group, co-founder of MEAALM Community, author of DevMagicFake Mocking Framework, Mohamed is a frequent [blogger](#) and speaker at Microsoft events.

How to Reduce Confusion and Rework Costs When Defining Requirements and Test Cases



Martin Crisp
CEO
[PowerStory](#)

1

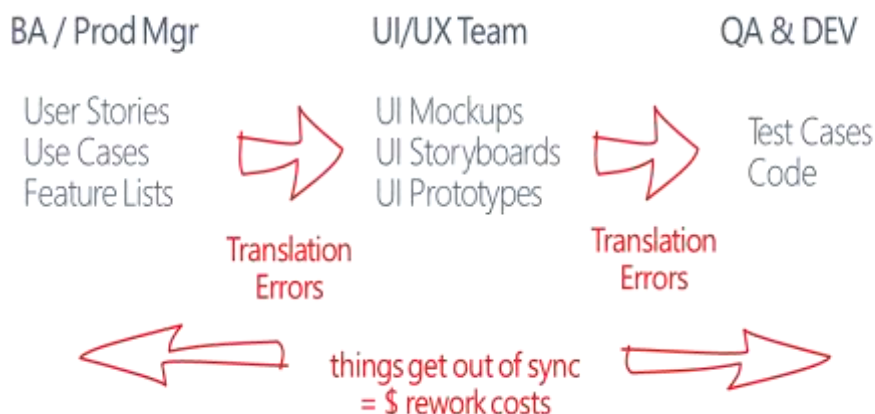
Current Approach

- Too Many Hand-offs And Deliverables

Many organizations approach to defining software requirements involve too many hand-offs and deliverables. This can lead to requirements translations errors between the different hand-offs and deliverables. In addition, the collective details of the requirements are spread across so many different deliverables it makes it harder for the developers and testers to properly interpret and understand the requirements.

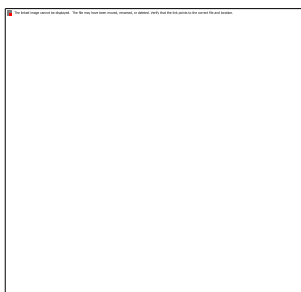
For example, many organizations will take this approach...

1. BA / Product Manager
 - The BA or Product owner elicits requirements from a customer...
 - ...and writes them down perhaps as a high level feature list...
 - ...then further elicits and documents requirements details using use cases and/or user stories and perhaps
2. UI / UX Team
 - These use cases or stories are then interpreted by the UI/UX team...
 - ...to develop UI Mockups
 - ...and perhaps UI Storyboards
 - ...or even full blow interactive UI Prototypes to bring these use cases to life.



Martin Crisp, Founder and CEO of [PowerStory](#), over the last 23 years has worked for large systems integrators, has founded and sold an early eCommerce startup, and has been CTO in multiple product software companies. Martin has a unique perspective on building both product and systems software using different techniques and tools for defining, communicating and validating software requirements.

Introducing Continuous Delivery In The Enterprise



Andrew Phillips
VP, Product Management
[XebiaLabs](#)

The Situation Today

Not too long ago customers and prospects had to find a computer to visit your site. In stark contrast with just a few years ago, this same customer is now online 24/7 via mobile devices and can be reached and can reach out continuously. The connection between customers and organizations is now fluid, and this phenomenon will only grow in the years to come.

Today, customers expect a reliable and completely up to date service anywhere, anytime. For your organization, this translates into constant pressure to add features and updates to your services almost instantaneously. This makes the quality of your online service more important than ever as issues with your online system would become visible in an instant. This also means that applications to support the business need to be deployed not only at a much faster rate, but also with more predictability.

For many organizations, a new approach to delivering software isn't simply about development and deployment practices; it's critical to doing business.

Challenges

In most organizations, the delivery of software is a time consuming, stressful and costly process. As soon as your application goes live, issues pop up, forcing the IT organization into another costly release cycle. A software release is typically a risky, unreliable procedure that costs businesses both time and expense. Transforming the release process into an automated, reliable, high quality continuous delivery pipeline that "just works" is critical to competing in today's marketplace. A process that enables the addition of new features to your online service in a matter of hours instead of days. Faster, better, cheaper..

[Andrew Phillips](#) is Vice President, Product Management, [XebiaLabs](#). XebiaLabs is a global provider of Application Release Automation and offers a Continuous Delivery platform to help enterprises improve and accelerate the application delivery process. To learn more about how to get started moving to Continuous Delivery, [Read the new white paper](#).

Using IIS Logs for Performance Testing with Visual Studio



Tarun Arora
Microsoft MVP
ALM Ranger
Consultant
[Avanade](#)

This will show you how you can play back the IIS Logs in Visual Studio to automatically generate the web performance tests. You can also [download](#) the sample solution demoed in this article.

Introduction

Performance testing is as important for new websites as it is for evolving websites. If you already have your website running in production you could mine the information available in IIS logs to analyze the dense zones (most used pages) and performance test those pages rather than wasting time testing & tuning the least used pages in your application.



What are IIS Logs

To help with server use and analysis, IIS is integrated with several types of log files. These log file formats provide information on a range of websites and specific statistics, including Internet Protocol (IP) addresses, user information and site visits as well as dates, times and queries. If you are using IIS 7 and above you will find the log files in the following directory `C:\Interpub\Logs\`

Walkthrough

1. Download and Install Log Parser from the Microsoft [download](#) Center.

[Tarun Arora](#) is a passionate technologist at [Avanade](#), London. Tarun is a Microsoft MVP and a Visual Studio ALM Ranger. After work he is found moderating MSDN forums, [blogging](#) and writing for MSDN, TechNet, Codeproject and other technology centric websites. For the latest and greatest follow him on [@arora_tarun](#)

Top 100 Agile Books (Edition 2012)



[Jurgen Appelo](#)

Author, Trainer, Coach



It took me four hours, three coffees, two toilet breaks, and one unpredictable wi-fi connection, but here it is... the annual **Top 100 Agile Books 2012**, based on ratings from *Amazon* and *GoodReads*.

There are some amazing changes at the top of the list this year.

Most notable is the new entry at #1 of **The Lean Startup**, by Eric Ries. Clearly deserved, as the whole Agile community has been talking about it all year.

But that's not all.

Specification by Example, by Gojko Adzic, makes a spectacular jump to the 2nd slot in the list. **The Scrum Field Guide**, by Mitch Lacey, has skyrocketed into the list at #7. And **The Culture Game**, by Dan Mezick, also makes a remarkable new entry at #39.

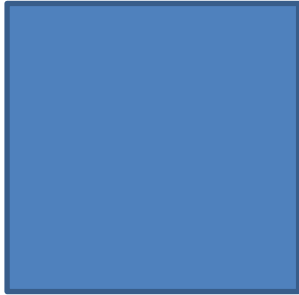
As for the rest, I suggest you see for yourself...

(For last year's list see: [Top 100 Agile Books, Edition 2011](#))

TY	LY	Title	Author(s)	Year
1	-	The Lean Startup	Eric Ries	2011
2	32	Specification by Example: How Successful Teams Deliver the Right Software	Gojko Adzic	2011
3	5	Succeeding with Agile: Software Development Using Scrum	Mike Cohn	2009
4	1	The Art of Unit Testing: With Examples in .Net	Roy Osherove	2009
5	9	The Agile Samurai: How Agile Masters Deliver Great Software	Jonathan Rasmusson	2010
6	2	Agile Estimating and Planning	Mike Cohn	2005
7	-	The Scrum Field Guide: Practical Advice for Your First Year	Mitch Lacey	2012
8	7	Agile Software Development, Principles, Patterns, and Practices	Robert C. Martin	2002

Jurgen Appelo is co-founder of the [Agile Lean Europe network](#) and the [Stoos Network](#) (focusing on change agents for organizational transformation). A [writer](#), [speaker](#), [trainer](#) who [blogs](#) on topics including Agile management, software engineering, business improvement, personal development, and complexity theory, Jurgen is the author of [Management 3.0: Leading Agile Developers, Developing Agile Leaders](#), describing the role of the manager in Agile organizations. He also wrote [How to Change the World](#), which describes his new supermodel for change management.

Integrating Development & Operations for the Enterprise



E

xecutive Overview
Organizations that

and fixes – and the organization benefits from that

Keith Pleas
Organizer and
Conference Chair for the
[ALM Summit](#)

¹ *hybrid IT* enables companies to use private clouds for application delivery and also take advantage of public cloud infrastructure to create resource flexibility and address temporary infrastructure demand needs

² Gartner Predicts 2012: Application Development (Dec 2011)

Interviews with Agile Evangelists



Derek Neighbors
[Integrum Technologies](#)

Agile Weekly PodCast

[Agile Manifesto with James Grenning](#)



*Click on the graphic above to play the interview
If the graphic above doesn't play for you
[click here for a dedicated player and the option to Download](#)*

Derek Neighbors and James Grenning discuss:

- History of The Agile Manifesto
- Embedded TDD
- Estimates

[Agile Weekly Podcasts](#)

[Derek Neighbors](#) is a serial entrepreneur who helps people bring ideas to reality. Derek co-founded [Gangplank](#), a collaborative workspace, in 2008 to encourage local "creatives" to explore innovative ideas. A partner at [Integrum Technologies](#), a consulting services firm, helping companies build high performing teams, Derek teaches entrepreneurship at [Arizona State University](#).

Embracing Modern App Development with Visual Studio 2012



Cheryl Hammond
ALM Consultant
[Northwest Cadence](#)

Whether it's Test-Driven Development, Test-First Development, or just plain figuring out whether your code does what you want it to do, unit testing is the standard for modern application development and a must-have for agile teams—but getting started can be a real challenge. What should you test? How granular should your tests be? How do you isolate code? How much code coverage is reasonable? If you've hit a wall, you're not alone.

Get straightforward, practical guidance you can use today to jump-start your team's unit testing efforts and finally see for yourself what the big deal is.

Streaming Video only available when connected to the Internet

When Online tap this screen icon to play the video
Once playing, tap the full screen icon and rotate your tablet



Click on the graphic above to play the video

If the graphic above doesn't play for you

[click here for a dedicated player and the option to Download](#)

James Tupper, ALM



[Northwest Cadence, Inc.](#) provides training and consulting services across the United States. The exclusive focus on Software Process and Microsoft ALM tools ensures clients benefit from deep technical expertise, sound process acumen, and hundreds of practical experiences on ALM projects. They partner with software development teams to improve processes and business results.

Visual Studio 2012 features enabled by using a TFS 2012 server



Buck Hodges
Director of Development
[Team Foundation Server](#)
[Microsoft](#)

While many of the features in Visual Studio 2012 are available to users using TFS 2008 or 2010, there are some features that are only available when using a TFS 2012 server. Also, the web experience in TFS 2012 has been rebuilt entirely from the ground up, and the result is a fast, fluid experience that also includes new experiences tailored to Scrum.

Once you've upgraded your server to TFS 2012, installed your first TFS server, or are using the [Team Foundation Service Preview](#), here are some of the features you are now able to use. The goal here isn't an exhaustive list but to get you started.

Version Control

- **Async checkout** – We have added a new TFS 2012 feature so that VS 2012 will do checkouts in the background for server workspaces. That eliminates the pause when you start typing and VS checks out the file. Turning it on turns off checkout locks, but you can still use checkin locks. You can find out how to turn it on [here](#).
- **Merge on Unshelve** – Shelvesets can now be unshelved into a workspace even if there are local changes on files in the shelveset. Conflicts will be created for any items modified both locally and in the shelveset, and you will resolve them as you would any other conflict.
- **Local workspaces** – [Local workspaces](#) allow many operations to be done offline (add, edit, rename, delete, undo, diff) and are recommended only *for workspaces with fewer 50,000 files*. Local workspaces are now the default with TFS 2012, but you can control that if you want server workspaces to be the default (setting is in the dialog shown [here](#)).

Alerts editor – Replacing the power tool, there's now an even better experience built into the web UI for managing your email notifications. You will be able to see and manage all of your alerts in one place. To get to it, go to a project, click on your name in the upper right, and choose Manage Alerts. Note that you can only get to it if your administrator has configured your server to send email.

Retry build – If your build fails for reasons unrelated to your changes, you can now right-click and retry it.

My Work – This is a new [feature](#) of Team Explorer that allows you to suspend and resume work quickly and easily. This feature is only available in Premium or Ultimate.

Code Review – You will be able to use the new code review experience. You can start a code review from the Pending Changes page in Team Explorer or from the History window by right clicking on a changeset. Your list of code reviews is shown in the My Work window. This feature also requires Premium or Ultimate.

Agile/Scrum – We've added a first-class experience for Scrum in TFS 2012 in our web UI. To use these, teams will need to be created. You can learn more [here](#).

- **Team** – We now have teams in TFS!
- **Task Board**
- **Product Backlog and Sprint Planning** – these require Premium or Ultimate (see this post about enabling via the [licensing feature](#))
- **Team Home Page** with a burn down chart and customizable tiles based on team favorites. To add tiles to a team's home page, create team favorites in work item tracking, build and version control.

Feedback – You'll be able to use this feature to request feedback on your work, and users will be able to use the [Microsoft Feedback Client](#) (including accessing it from [Microsoft Test Manager](#)).

Storyboarding – This is available in VS Premium or Ultimate and in Test Pro, and TFS 2012 adds a Storyboard tab to work items such as User Story. You can learn more [here](#) and find additional shapes [here](#).

You can also use the [new unit testing features](#) to run tests via Nunit, xunit, qunit, and more in your TFS build (aka team build).

Microsoft Test Manager – there are a few features only available when using MTM 2012 with TFS 2012

- Inline Reports/Test Results (Testing Center -> Plan Activity -> Results)



- Exploratory feature (Testing Center->Test Activity->Do Exploratory Testing, Testing Center->Test Activity->View Exploratory Test Sessions)

Do Exploratory Testing

To begin exploratory testing without any selected work items: [Explore](#)

Perform tests using: Local machine (RAVIDEV11FB) [Modify](#)

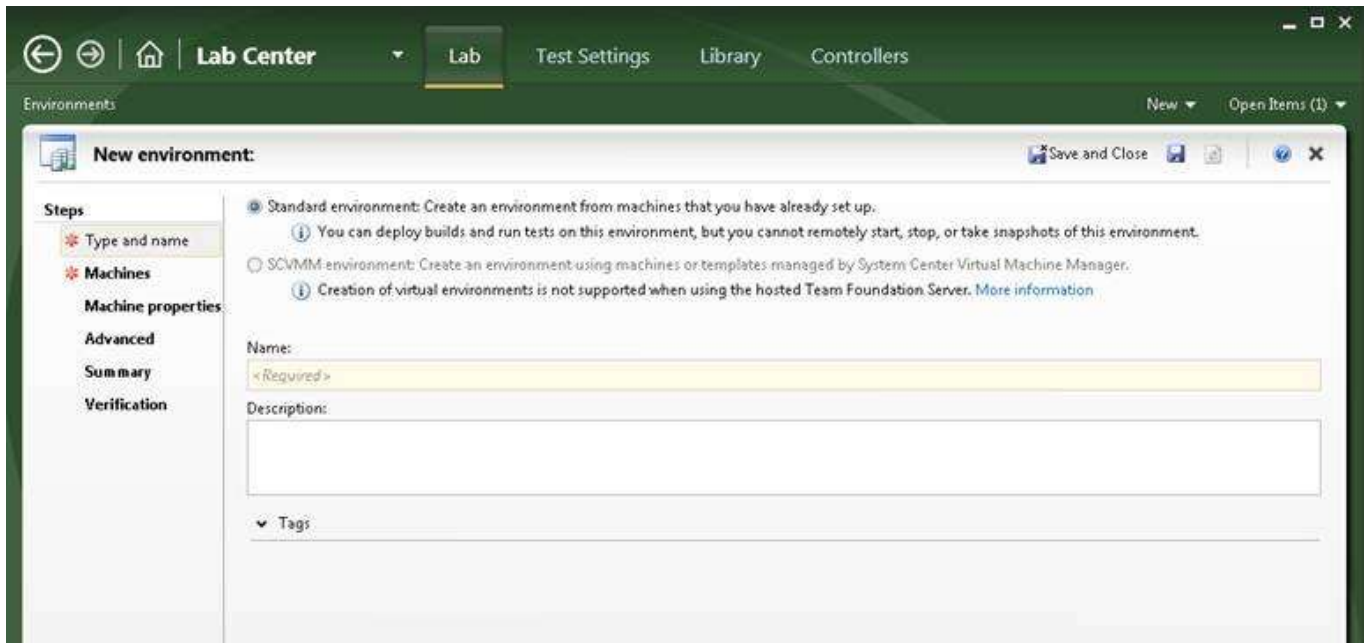
To test specific work item(s), select work item(s) below

Explore work item [Open](#) [Unfiltered](#)

Drag a column header here to group by that column.

ID	Title	Assigned To	State	Area Path
16	As a user, I should be able to rent movies onl...		New	scrumproj
17	As a manager, I want to create/edit users of t...		Done	scrumproj
18	As a manager, I want to delete a user		Done	scrumproj
19	As a customer, I want to register myself		Done	scrumproj
20	As a customer, I want to edit my profile		Done	scrumproj
21	As a customer, I want to cancel my members...		Done	scrumproj

- Support for Standard Environments (Lab Center->Lab Activity->Environments)



[TFS Express](#) does not include any of the Agile features. It is really focused on source control, build, and bug tracking. You can read more about it [here](#)

For a more detailed list of the features in ALM for 2012, see the [ALM 2012 section](#) or download the [product guide](#). There's also a list of [what's new in VS 2012](#).

[Buck Hodges](#) is director of development for [Team Foundation Server](#) at [Microsoft](#). Since 2003 Buck has worked directly on most aspects of all versions of TFS including: Team Foundation Build, Team Web Access and Team Foundation Server Version Control contributing mostly to the version control object model, the `tf.exe` command line, and general TFS infrastructure. Twitter: [@tfsbuck](#).



POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with Xebialabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

Xebialabs
the deployment automation company

Discover the Xebialabs
Continuous Delivery Platform.
www.xebialabs.com

next thing. The time box limits the amount of work a team can take on.

[Tobias Mayer](#) is a facilitator, guide, and coach, specializing in Agile transformation. His current work explores the concept of [craftsmanship](#) in this context, focusing on the values and principles that foster an Agile mindset.

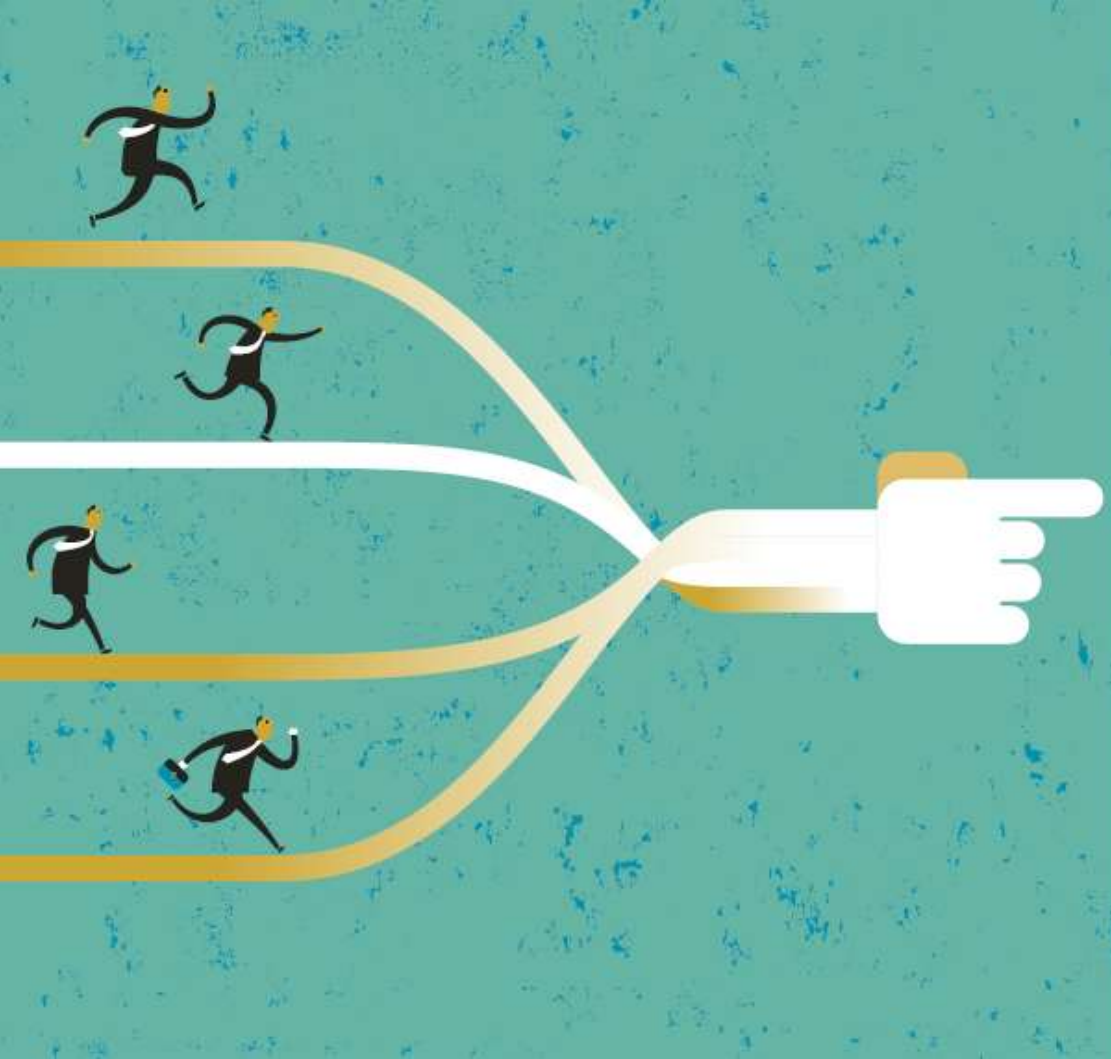


POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with Xebialabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

Xebialabs
the deployment automation company

Discover the Xebialabs
Continuous Delivery Platform.
www.xebialabs.com



POWER your DevOps & Release Teams

Automatically deploy .NET applications and configuration to middleware and cloud environments with Xebialabs' Deployit. Fast. Reliable. Cost Effective deployments with TFS integration and Auto-rollback — even in complex multi-tier environments.

Xebialabs
the deployment automation company

Discover the Xebialabs
Continuous Delivery Platform.
www.xebialabs.com

correctly.

[Andrew Phillips](#) is Vice President, Product Management, [Xebialabs](#). Xebialabs is a global provider of Application Release Automation and offers a Continuous Delivery platform to help enterprises improve and accelerate the application delivery process. To learn more about how to get started moving to Continuous Delivery, [Read the new white paper](#).

Micros



Tomáš Tureček
Lean/Agile Coach
[Tieto Corporation](#)

Welcome to

In this article I'll introduce you to the new Microsoft Certified Solution Developer: Application Lifecycle Management certification and then focus on the

[Tomáš Tureček](#) PhD, an Lean/Agile coach in [Tieto Corporation](#), has been working in IT industry for more than 15 years in various roles from development to management. He has more than 6 years of experience from coaching and mentoring deliveries in distributed environments. Tomáš contributes to IT communities, [blogs here](#) and speaks at prestigious international conferences.

1: The Old Way and the New Way

Today, software must meet your customers' needs in an ever-changing landscape. New features must be released continuously, updates must be timely, and bug fixes cannot wait for version 2. That's the reality in software development today, and it is never going back to the days of a new release every few years. That's especially true for cloud applications, but it's true for software in general as well.

*Howie Hilliker
Microsoft*

Such agility requires lifecycle management that's built specifically to meet modern needs. But there are obstacles. Testing is one aspect of software development that can present a roadblock to agility. To publish even a one-line fix, you typically need to re-test the whole system. That can mean finding and wiring up the hardware, reassembling the team of testers, reinstalling the right operating systems and databases, and setting up the test harness. In the past, this overhead prevented the continuous update cycle that is so necessary today.

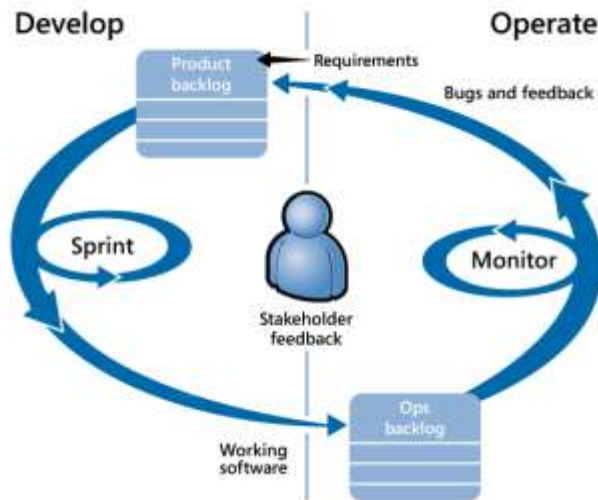
Many teams are still operating that old way, putting their competitiveness at risk. Fortunately, yours does not have to be such a team. You don't have to build and test software the way your ancestors did.

So how can you move your software testing into the 21st century? That is the subject of this book. This book is about how to streamline the testing of your software so that you can make updates and fix bugs more rapidly, and continuously deliver better software.

Because testing can be a significant obstacle, our aim in this book is to help you substantially reduce the overhead of testing while improving the reliability and repeatability of your tests. The result should be a shorter cycle of recognizing the need for a change, making the change, performing the tests, and publishing the update.

The planning and execution of this cycle is known as application lifecycle management (ALM).

Tobias Mayer



The application lifecycle

Application lifecycle management with Visual Studio

Microsoft Visual Studio will figure prominently in this book and in your efforts to refine your own application lifecycle management. Application lifecycle management with Visual Studio is an approach that takes advantage of a number of Microsoft tools, most of which are found in Visual Studio and Team Foundation Server, and which support each part of the lifecycle management process. In these pages, we'll show you how to perform testing with Visual Studio Team Foundation Server 2012 RC. Though we will focus on that version, you can also use Visual Studio 2010. (We note the differences where necessary.) With the setup guidance we provide here, your team should be able to test complex distributed systems within a few days.

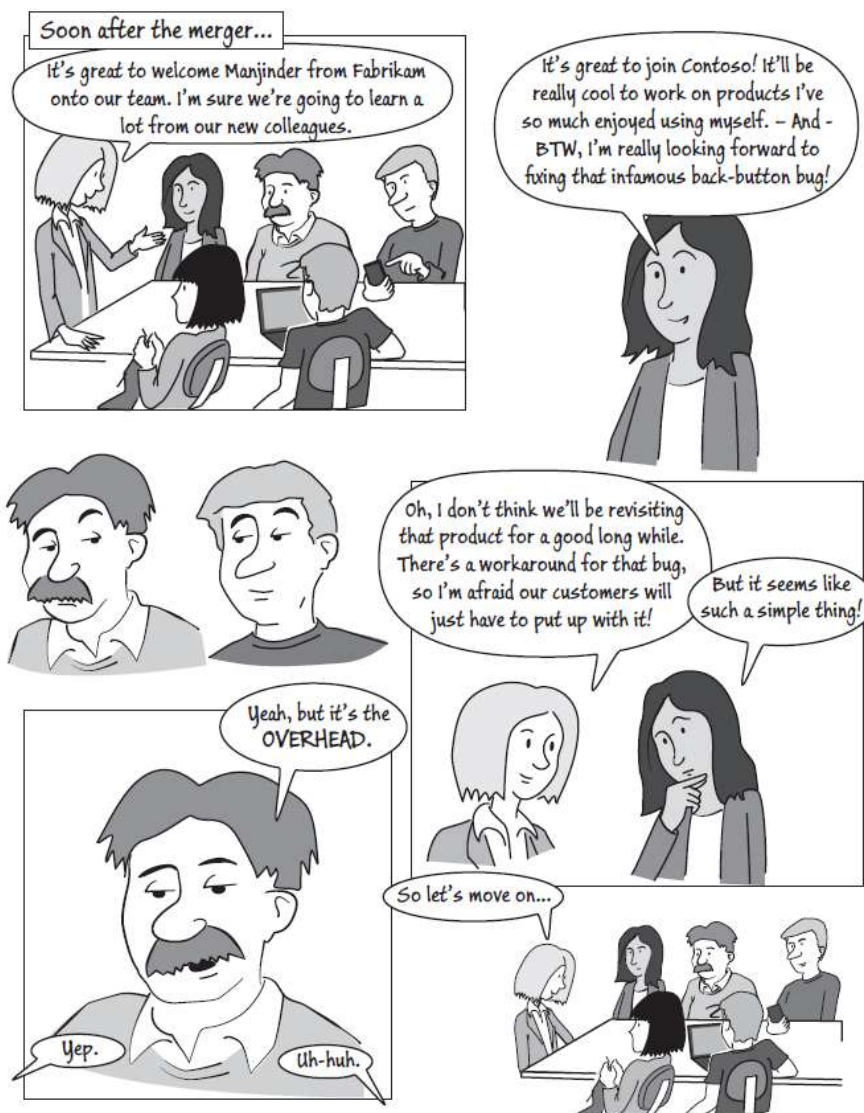
Of turnaround. If your software runs in the cloud—that is, if users access it on the web—then you will course, you may already be doing everything we suggest here, so we can't promise the scale of improvements you'll see. But we envisage that, like us, you're concerned with the increasing demand for rapid probably want no more than a few days to pass between a bug report and its fix. The same is true of many in-house applications; for example, many financial traders expect tools to be updated within the hour of making a request. Even if your software is a desktop application or a phone app, you'll want to publish regular updates.

Before we dig into lifecycle management with Visual Studio, let's take a look at the two fictitious companies that will introduce us to two rather different approaches to testing and deployment: Contoso, the traditional organization, and Fabrikam, the more modern company. They will help define the ALM problem with greater clarity.

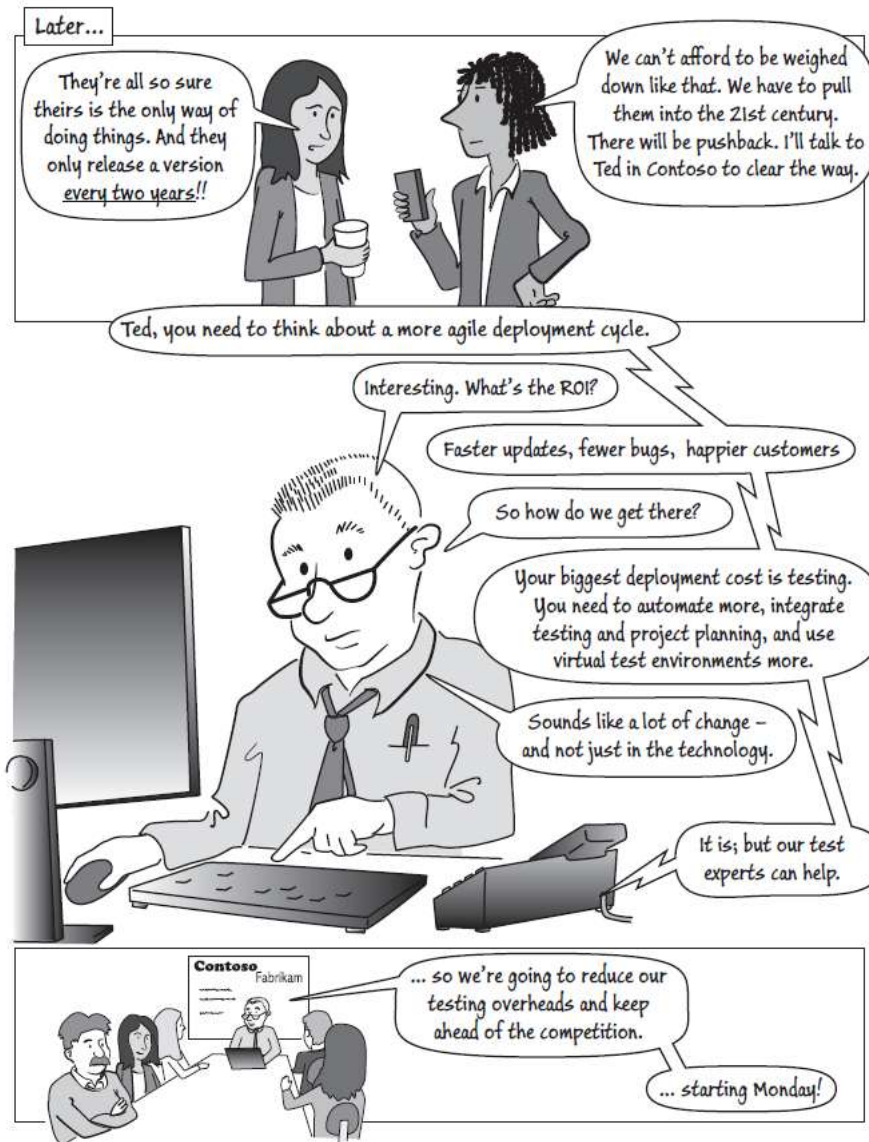
Contoso and Fabrikam; or something old, something new

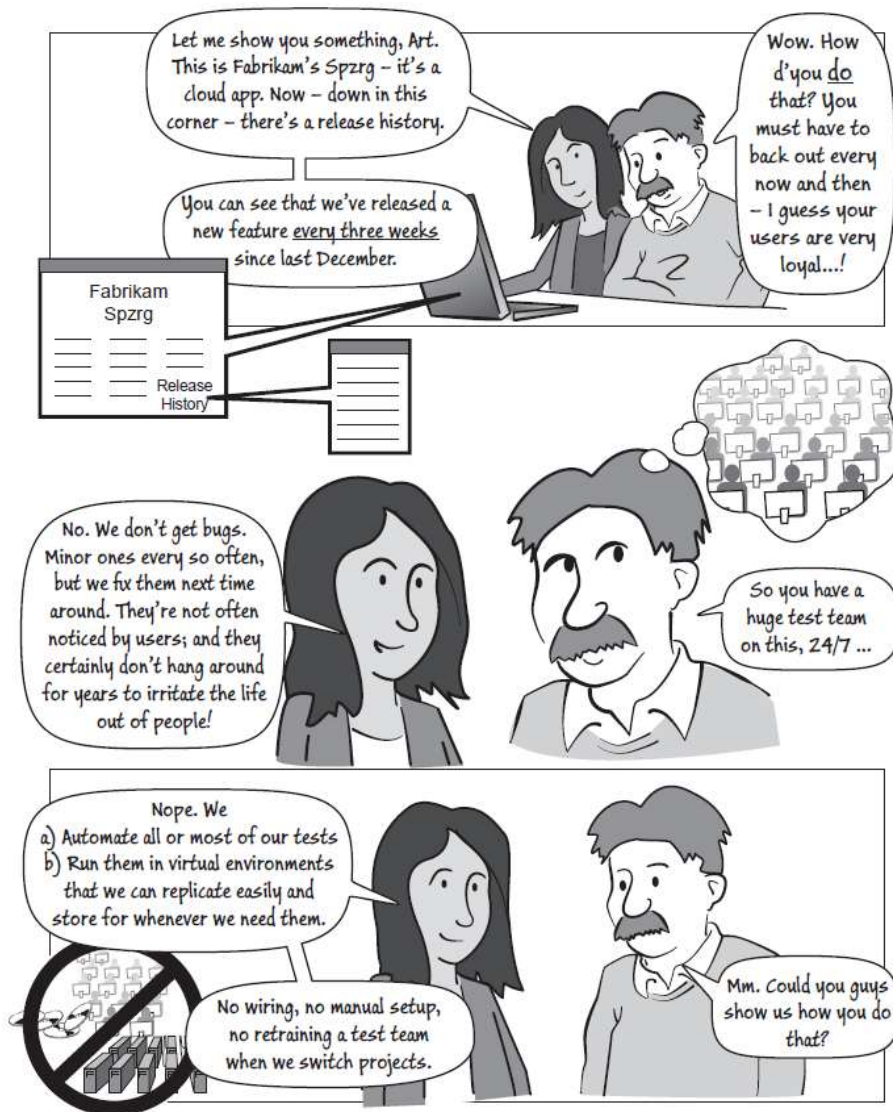
At Contoso, testing has always been high priority. Contoso has been known for the quality of its software products for several decades. Recently however, testing has begun to seem like a dead weight that is holding the company back. When customers report bugs to Contoso, product managers will typically remark "Well, the fix is easy to code, but we'd have to re-test the whole product. We can't do that for just one bug." So they have to politely thank the customer for her feedback and suggest a workaround.

Meanwhile, their competitor, Fabrikam, a much younger company with more up-to-date methods, frequently releases updates, often without their users noticing. Bugs are gone almost as soon as they are reported. How do they do it? How do they test their whole product in such a short time? In our story, the problems begin when Fabrikam and Contoso merge. The two cultures have a lot to learn from each other.









The two companies take very different approaches, and yours might be somewhere between the extremes. In the chapters that follow, we show you how to get set up to do things the new way, and we explain the choices you can make based on the situation in your organization.

From Contoso to Fabrikam

Let's take a look at the pain that Contoso experiences in the beginning and the benefits they realize as they move to a testing process more like Fabrikam's. You'll get a better understanding of the benefits to testing using Visual Studio, Team Foundation Server, and a virtual lab environment. Here are some of the pain points that Contoso experienced by doing things the old way:

- Updating an existing product is expensive no matter how small the change. Partly, this is because test hardware has to be assigned, private networks have to be wired up, and operating system and other platform software has to be installed. This is a lengthy process. As a result, if customers find bugs in the operational product, they don't expect to see them fixed soon.

- During a project, the team frequently needs to test a new build. But the results are sometimes inconsistent because the previous build didn't uninstall properly. The most reliable solution would be to format the disk and install everything from scratch; but this is too costly.
- Manual tests sometimes yield inconsistent results. The same feature is tested by slightly different procedures on different occasions, even if the same person is doing the test.
- Developers often complain they can't reproduce a bug that was seen during a test run. Even if the tester has faithfully reported the steps that reproduce the bug, the conditions on a development machine might be different from those on the test environment.
- When the requirements change, it can be difficult to discover which tests should be updated; and it can be difficult to find out how well the latest build of the product meets the stakeholders' needs. Different unintegrated tools mean that there are no traceable relationships between bugs, tests, customer requirements, and versions of the product. The manual steps needed to make the tools work together make the process unreliable.
- Testing at the system level is always manual. To repeat a set of tests is costly, so that it is not economical to develop the software by incremental improvements. System testing is often abbreviated, so that bugs can go undiscovered until release.
- Revisiting code is risky and expensive. Changing code that has already been developed means rerunning tests, or running the risk that the changes have introduced new bugs. Development managers prefer to develop all of one part of the code, and then another, integrating the parts only towards the end of the project when it is often too late to fix integration problems.

Here are the benefits Fabrikam enjoys and which Contoso will realize by moving to the new way:

- Virtual machines are used to perform most tests, so new testing environments can be easily created and existing environments can rapidly be reset to a fresh state.
- Configurations of machines and software can be saved for future use: no more painful rebuilding process.
- Manual tests are guided by scripts displayed at the side of the screen while the tester works. This makes the tests more repeatable and reliable. Test plans and test scripts are stored in the Team Foundation Server database, and linked to the code and to the test configuration.
- Tests are easy to reproduce. Testers' comments, screenshots and actions can be recorded along with the test results. When a bug is found, a snapshot of the exact state of the environment is stored along with the bug. Developers can log in to the environment to investigate.

- Much more test automation. Manual test steps can be recorded and replayed rapidly. The recordings can also form the basis of automated system tests. Automated tests can be performed frequently, and at little cost.
- Reports on the project website show the progress of requirements in terms of the relevant tests.
- When requirements change, it's easy to trace which tests should be updated. An integrated set of tools manages and reports on requirements, test cases and test results, project planning, and bug management.
- In addition to the core tools, third-party products can be integrated.
- Changing existing code is commonplace. Automated tests are performed frequently and can be relied on to pick up bugs that were inadvertently introduced during a change. Development teams can produce a very basic version of an end-to-end working product at an early stage, and then gradually improve the features. This substantially improves the chances of delivering an effective product.

Application lifecycle management tools

We'll assume that you've met Visual Studio, the Microsoft software development environment. As well as editing code in Visual Studio, you can run unit tests, break into the code to debug it, and use the IntelliTrace feature of Visual Studio to trace calls within the running code. You can analyze existing code with dependency, sequence, and class diagrams, and create models to help design or generate code.

Team Foundation Server is a tool for tracking and reporting the progress of your project's work. It can also be the source control server where your developers keep their code. It can build and test your software frequently as it grows, and provide reports and dashboards that show progress. In particular, you can get reports that show how far you've gone towards meeting the requirements—both in terms of work completed and tests passing.

Microsoft Test Manager (MTM) is the tool for testers. With it, you can plan and execute both manual and automated tests. While you are performing tests, you can log a bug with one click; the bug report will contain a trace of your recent actions, a snapshot of the state of the system, and a copy of any notes you made while exploring the system. You can record your actions in the test case, so that they can be played back on later occasions.

MTM also includes tools for setting up and managing lab machines. You can configure a virtual lab in which to install a distributed system, and link that lab to the test plan. Whenever you need to repeat tests—for example when you want to publish a change to your system—the lab can be reconfigured automatically.

Moving to the new way: adopting Visual Studio for ALM

Now we aren't suggesting that everyone should work by the same methods. Some teams want to aim for a rapid development cycle; others are developing embedded software for which that wouldn't be appropriate.

Nor do we suggest that people in Contoso are doing it all wrong. Their attitude to testing is clearly admirable: they take a pride in releasing high-quality software.

But we do believe that, no matter what development books a team reads, they do need to run tests, and they can benefit from having a brisker turnaround in their test runs—if only to reduce the boredom of repeating the same old tests by hand.

But such changes aren't just about adopting tools. A software team—the extended team that includes all the stakeholders who collaborate to produce the working software—consists of interacting individuals. To keep everyone in sync, new ways of doing things have to be tried out and agreed upon in measured steps.

But such changes aren't just about adopting tools. A software team—the extended team that includes all the stakeholders who collaborate to produce the working software—consists of interacting individuals. To keep everyone in sync, new ways of doing things have to be tried out and agreed upon in measured steps.

Visual Studio provides a *lot* of different facilities. It therefore makes sense to adopt it one step at a time. Of course, this book is about testing, but since the test tools and other features such as work tracking and source control are closely integrated, we have to talk about them to some extent too.

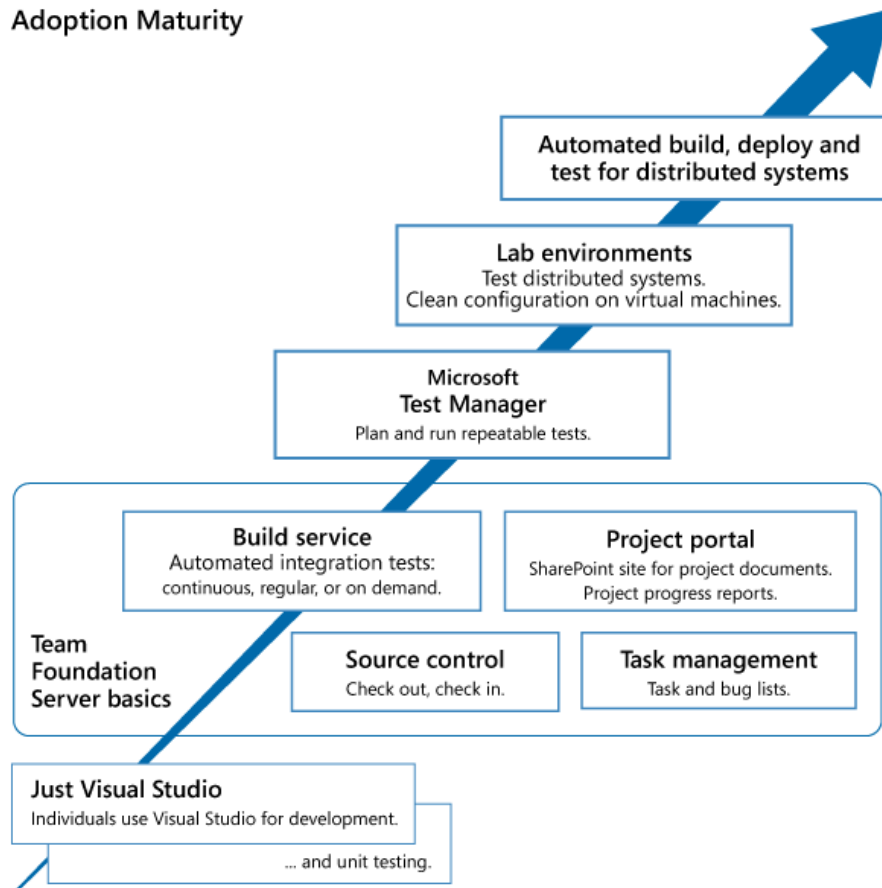
The following diagram shows a typical order in which teams adopt Visual Studio for application lifecycle management. They begin by just using Visual Studio, and work gradually on up through source control, server builds, system testing, and on to automated system tests. The need for gradual progress is mostly about learning. As your team starts to use each feature, you'll work out how to use it in the best way for your project.

And of course a team learns more slowly than any of its members. Learning how to open, assign, and close a bug is easy; agreeing who should do what takes longer.

Adopting Visual Studio for ALM

There are a number of ways you can gradually adopt Visual Studio for ALM, and the steps below represent one way. Naturally, this isn't a precise scheme. But here's what you get in each of the stages of adoption that we've shown:

Adoption Maturity



- **Just Visual Studio** – Visual Studio is used for development. To test the application, Contoso's testers basically just press F5 and find out whether the application works. Unit tests are written by the developers to test individual classes and components. Coded UI Tests are a neat way to run automated tests of the whole application through its user interface.
- **Team Foundation Server Basics** – when you install Team Foundation Server, you get a host of features. The first features you'll want to take advantage of are:
 - **Source Control** to avoid overwriting each other's work. And after a while, you might start using:
 - *Check-in rules* – which remind developers to run tests and quality analysis on their code before checking it into the server.
 - *Shelvesets* – a way of copying a set of changes from one user to another for review before checking in.

- *Branches* - which help manage work in a large project.
- **Task Management** is about tracking the backlog (the list of tasks), bugs, issues, and requirements. Each item is recorded in a *work item*. You can assign work items to people, project iterations, and areas of work; you can organize them into hierarchies; and you can edit them in Excel, and sync with other project management tools.
- The **Project Portal** is a SharePoint website, integrated with Team Foundation Server so that each project automatically gets its own site. And, even more interestingly, you can get some very nice dashboards, including graphs and charts of your project's progress. These reports are based on the work items, and on the test results.
- The **Build Service** is a feature of Team Foundation Server that performs a vital function for the development team. It builds all the code that has been checked in by developers, and runs tests. Builds can run on a regular or continuous cycle, or on demand. The team gets email alerts if a compilation or test fails, and the project portal shows reports of the latest results.

The email alert is very effective at keeping code quality high: it prominently mentions who checked in code before the failed build.

- **Microsoft Test Manager** is where it gets interesting from the point of view of the professional tester. Microsoft Test Manager makes tests reliably repeatable and speeds up testing. Using it, you can:
 - Write a script for each manual test, which is displayed at the side of the screen while the test is being performed.
 - Partially automate a test by recording the test actions as you perform them. The next time you run the test, you can replay the actions.
 - Fully automate a test so that it can run in the build service. To do this, you can adapt code generated from partly automated tests.
 - Associate tests with user requirements. The project portal will include charts that display the progress of each user requirement in terms of its tests.
 - Organize your tests into suites and plans, and divide them up by functional areas and project iterations.
 - Perform one-click bug reporting, which includes snapshots of the state of the machine.

- **Lab Environments** are collections of test machines—particularly virtual machines. Without a lab, you can test an application locally, running it on your own computer. During development, applications are typically debugged on the development machine, often with several tiers running on the same machine. But with lab facilities, you can:
 - Deploy a system to one or more machines and collect test data from each machine. For example, a web client, Microsoft Internet Information Services (IIS), and a database would run on separate machines.
 - Run on freshly-created virtual machines, so that there's no need to uninstall old versions, no chance of the application corrupting your own computer, and you can choose any platform configuration you like.
 - Configure an environment of virtual machines for a particular test suite, and store it for use whenever you want to run that suite again.
 - Take a snapshot of the state of an environment and save it along with a bug report.
- **Automated build, deploy, and test.** The simplest setup of the build service runs unit tests in the same way the developer typically does—all on one machine. But for web and other distributed applications, this doesn't properly simulate the real operational conditions. With automated deployment, you can run tests on a lab environment as part of the continuous or regular build. The automation builds the system, instantiates the appropriate virtual environment for the tests, deploys each component to the correct machine in the environment, runs the tests, collects data from each machine, and logs the results for reporting on the project portal.

Click here to read the remaining chapters and sections:

<http://msdn.microsoft.com/en-us/library/jj159345.aspx>

- **Chapter 2: Unit Testing: Testing the Inside**
- **Chapter 3: Lab Environments**
- **Chapter 4: Manual System Tests**
- **Chapter 5: Automated System Tests**
- **Chapter 6: A Testing Toolbox**
- **Chapter 7: Testing in the Software Lifecycle**
- **Appendix: Setting up the Infrastructure**
- **The development process**
- **Testers vs. developers?**
- **Agile development**

<http://msdn.microsoft.com/en-us/library/jj159336.aspx>

The revision history is atomically stored for each change but also allows for merging as long as there

Figure: Customize Work Item Types

Because everyone is authenticated the same across all parts of the system you have a single model for

process helps organizations manage and

used in development and test labs. New tools

These automated procedures, called

Integrated production monitoring and diagnostics

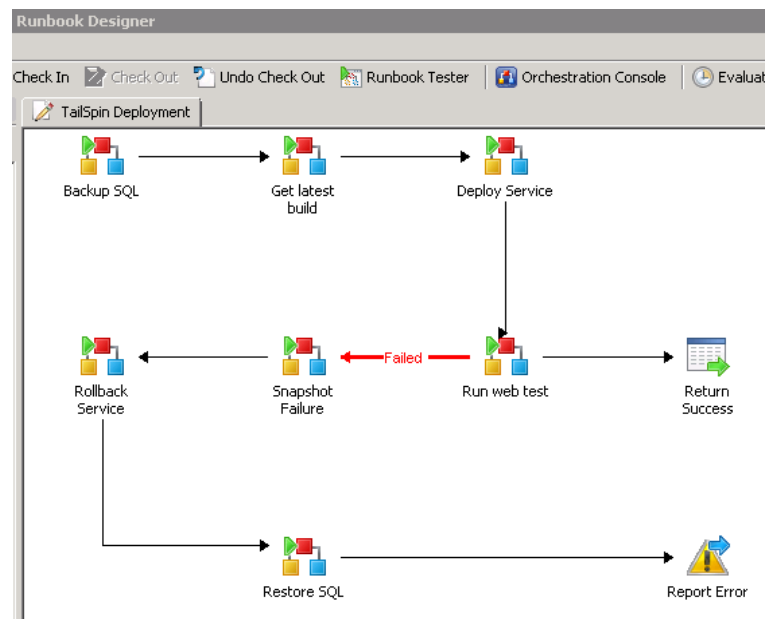


Figure 1: Automated Procedures

Finally, TFS has been extended to enable

performance from every angle with a minimal

problems), functionality (is the app doing what

System Center 2012 introduced a new cloud

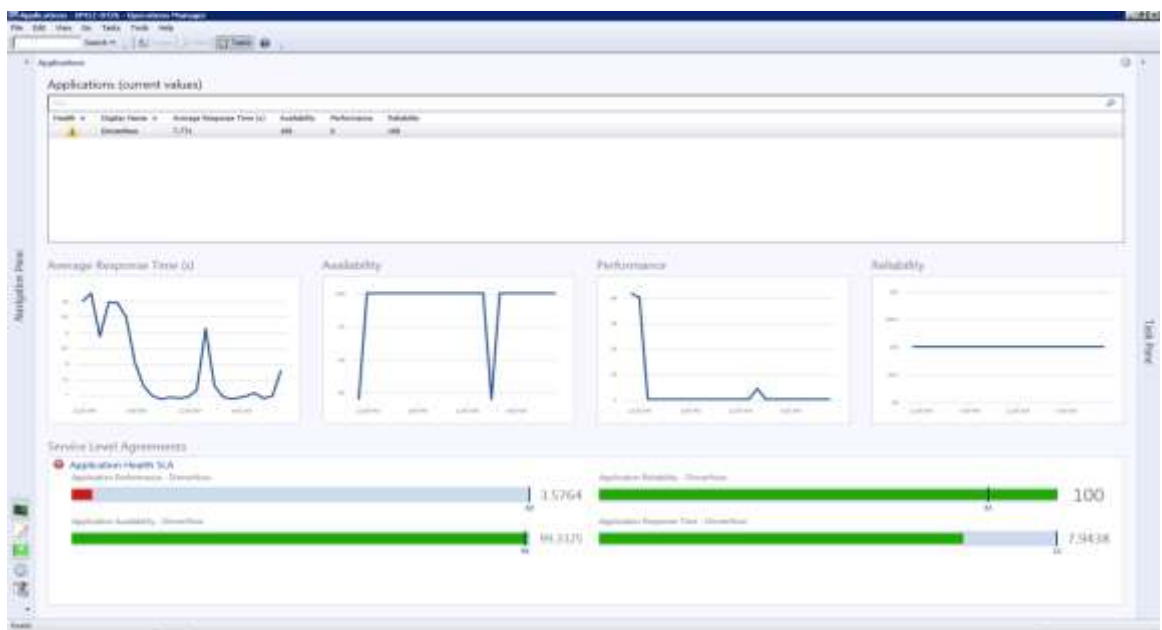
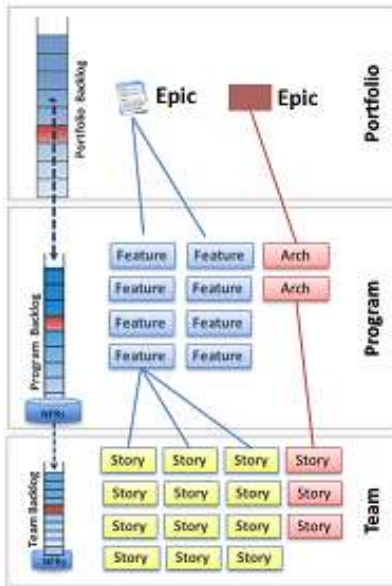


Figure 2: Application Dashboard

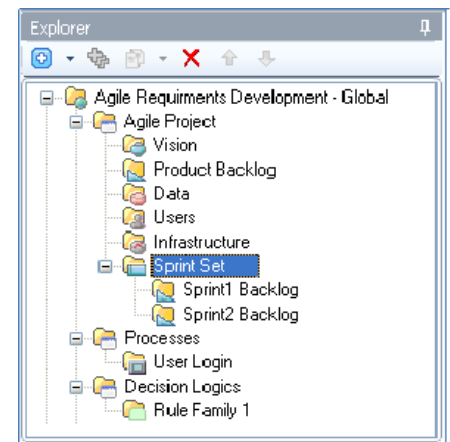
⁵ *Multi-tenant* software runs on one server, but services multiple client organizations (tenants)

In the Scaled Agile Framework, a Product Backlog is considered as an



Leffingwell has scaled iterations into multiple levels in his Scaled Agile Framework. At the Team level,

In inteGREAT, a flexible framework is provided to support Agile



[Bob Savelson](#), a seasoned veteran of the software development industry, has worked for large outsourcers and built his own requirements facilitation company; helping many clients to optimize their ALM processes and define the 'right' requirements. Today, as SVP at [eDevTech](#), he continues to promote [inteGREAT](#) and [SmartOffice](#) as industry leading requirements tools.