

ALM Mag.com

Empowering Rapid Delivery of Quality Software

**Interview – Brian Harry
Discusses ALM & TFS**



**The Principles
of Agile Architecture**

**Deliver Early, Often,
And Exceptionally**

***Top 100*
Agile Books**

**Continuous Delivery
In The Enterprise**

**TF.exe Fast
Overview**

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

Microsoft
ALM
Certification
– Part 1 of 3**

**Using IIS Logs for
Performance Testing**

Featured Article

Interview –

***Brian Harry Discusses
the Farm, ALM and TFS***

With Focus on TFS Update 2



13

Features

04

Welcome to ALM Mag

Editor's Welcome



05

The Principles of Agile Architecture

Proven patterns to more rapidly evolve current systems

by Alex Yakyma and Dean Leffingwell



12

***Deliver Early, Often, and Exceptionally
with VS Premium***

Speed the delivery of high-quality, on-target applications

by Cheryl Hammond



- 25** ***Microsoft ALM Certification – Part 1 of 3***
Learn about Microsoft's new ALM certification
by Anthony Borton 
- 29** ***TF.exe Fast Overview***
A fast overview covering the tf.exe command line
by Mohamed Radwan 
- 35** ***How to Reduce Confusion and Rework Costs
When Defining Requirements and Test Cases***
An approach that helps reduce project costs and increase the likelihood
that you will develop what your customers need
by Martin Crisp 
- 41** ***Introducing Continuous Delivery
In The Enterprise***
Streamlining all parties involved: from business all the way to operations
by Andrew Phillips 
- 53** ***Using IIS Logs for Performance Testing
with Visual Studio***
How to play back the IIS Logs in Visual Studio to automatically generate
the web performance tests
by Tarun Arora 
- 59** ***Top 100 Agile Books (Edition 2012)***
Top 100 Agile Books 2012, based on ratings from Amazon and
GoodReads
by Jurgen Appelo 

Welcome



Keith Denham
Publisher

Treasure Cove Publishing
Volume 1 – Issue 3
March/April 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 features an interview with Brian Harry, Technical Fellow at Microsoft and a leader of the TFS team. Brian discusses some of the new features included in the Visual Studio 2012 Update 2 CTP.

If this was a physical magazine you'd be noticing that it is increasing in thickness with each issue even though this issue moves to a bi-monthly format to accommodate an increased staff engagement schedule. This month although there is on central theme, we have another great lineup.

Alex Yakyma and Dean Leffingwell outline the *Seven Principles of Agile Architecture* and the benefits of their application when building enterprise class software architectures in a Lean and Agile fashion.

Cheryl Hammond presents another great Northwest Cadence video demo presentation on how the use of TFS 2012 can support an increased quality and delivery cadence.

Anthony Borton article is the first of a three part series outlining the new Microsoft MCSD ALM certification. Part 1 introduces the new certification and covers 70-496: Administering Microsoft Visual Studio Team Foundation Server 2012.

Martin Crisp offers insights into how to gain clarity and reduce errors thorough the integration of an Agile User Story, automated Test Case TFS tool. Mohamed Radwan and Tarun Arora cover a couple of interesting TFS features with some great examples.

Andrew Stevens of XebiaLabs follows his article in the February issue by discussing Continuous Deployment concepts in more depth.

For those in an Agile frame of mind, don't miss Jurgen Appelo's annual evaluation of the Top 100 Agile Books available and their ratings.

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

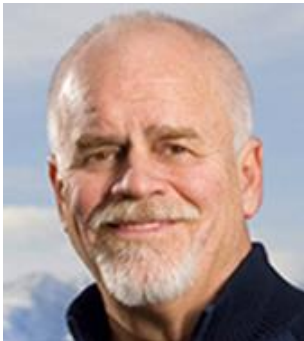
Keith Denham

*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.*

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.

Details

In the text *Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise* [1], we introduced Eight Principles of Agile Architecture to provide guidance to the development of enterprise class systems in a Lean and Agile manner. Now, a few years later, we have the experience in applying these principles, to largely good effect, in a significant number of large enterprises adopting SAFe. Through use, we've found it helpful to revise and reorder them to make things clearer, slightly more prescriptive, and to make the story telling a

little easier in our courseware and consulting practice. In this next section, we'll describe the *Seven Principles of Agile Architecture* that we now apply to address the challenges of building enterprise class software architectures in a Lean and Agile fashion:

1. Design emerges. Architecture is a collaboration.
2. The bigger the system, the longer the runway
3. Build the simplest architecture that can possibly work
4. When in doubt, code, or model it out
5. They build it, they test it
6. There is no monopoly on innovation
7. Implement architectural flow

Principle 1 – Design Emerges. Architecture is a collaboration

Traditional, waterfall development methodologies used what we have come to call Big Up-Front Design (BUFD) to create a roadmap and architectural infrastructure for our future system. Our hope was that BUFD would capture requirements and architectural plans sufficiently adequate to support the system for years to come. And perhaps they did. However, we also discovered considerable challenge when it inevitably came to adapting “a massive set of speculative, forward looking constructs” to the actuality of the real world. Soon enough, the designs were brittle and hard to change, and eventually a big-branch-and-merge to a new set of speculative assumptions was the routine course of action.

Emergent Design

Agile development eschews waterfall thinking and BUFD and replaces it with a simple belief “*the best architectures, requirements, and designs emerge from self-organizing teams*” (*Agile Manifesto* [2]). Out of this comes the practice of *emergent design*—the evolutionary process of discovering and extending the design only as necessary to implement and validate the next increment of functionality. Teams used their new coding languages, testing tools, and refactoring skills to rapidly evolve the design in accordance with the then-current requirements. This new practice works extremely well, up to a point.

Intentional Architecture

As Agile practices matured and became adopted by larger teams and teams of teams, however, there comes a point at which *emergent design* is an insufficient response to the complexity of large-scale system development. Simply, it is not possible for teams to anticipate changes that may well occur outside their environment, nor for individual teams to fully understand the entire system and thereby avoid producing redundant, and/or conflicting code and designs. Simply put, no one team in a larger software enterprise can see the bigger picture, nor reasonably anticipate some of the changes that are headed their way, many of which arise outside their local control. For this we need some *Intentional Architecture*—a set of purposeful, planned architectural initiatives to enhance solution design, performance and usability—and which provides guidance for inter-team design and implementation synchronization.

Collaboration

Clearly, in the context of Agile enterprise software systems, we need both: fast, *local control of emergent design* so that teams react appropriately to changing requirements without excessive attempts to future proof the system, and *global control of Intentional Architecture*, the guidance needed to assure that the system as a whole has conceptual integrity and efficacy. *Achieving the right balance of emergent design and intentional architecture drives effective evolution of the system*, as Figure 1 illustrates.

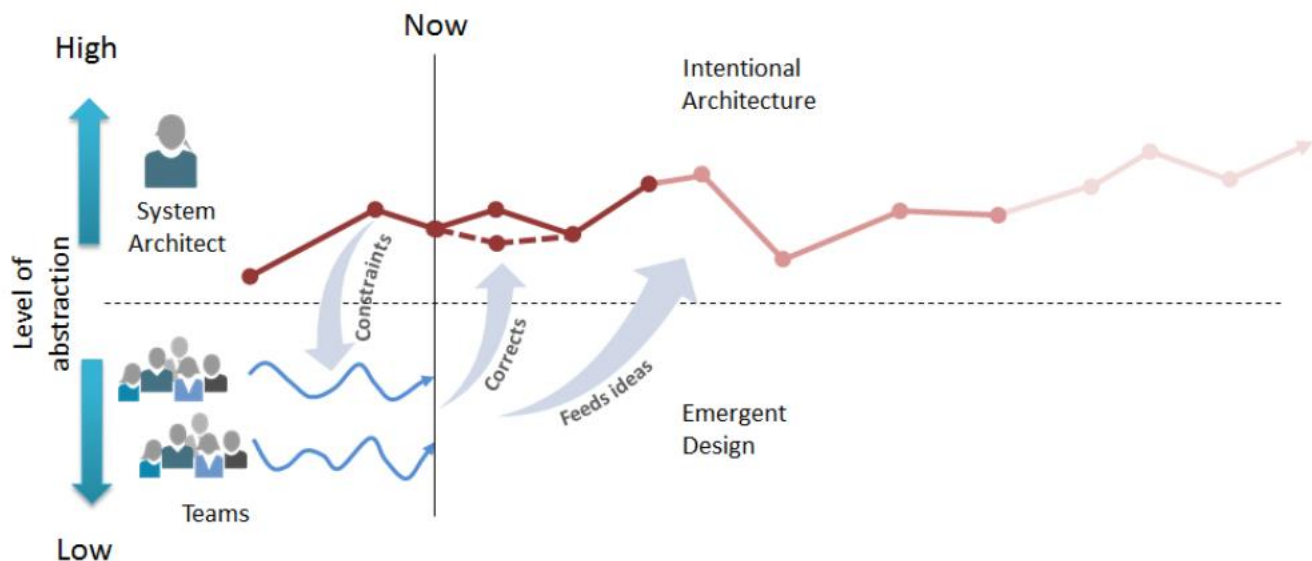


Figure 1. Intentional architecture, emergent design and collaboration support system evolution

Moreover, they are not independent. Intentional architecture constraints the emergent design, but at a high enough level of abstraction to allow the teams to effectively adapt the “intentional” part to their specific context. At the same time, emergent design influences and corrects intentional architecture, and also feeds new ideas for future, centralized, intentional effort.

Such a deep reciprocity between emergent design and intentional architecture can occur only as a result of *collaboration* between [Agile Teams](#), [System](#) and [Enterprise](#) Architects, [Product Management](#) and even [Program Portfolio Management](#) in leveraging architecture to optimize business performance. In SAFe, one primary construct, the self-organizing team-of-Agile-teams we call the [Agile Release Train](#), plays a key role in fostering this collaboration and building effective solutions.

Principle 2 – The bigger the system the longer the runway

In SAFe, [Architectural Runway](#) exists when the enterprise’s platforms have sufficient technological infrastructure to support the implementation of the highest priority epics and features in the backlog without excessive, delay-inducing, redesign. In order to achieve some degree of runway, the enterprise must continually invest in refactoring and extending existing platforms as well as building and deploying those new platforms needed for evolving business requirements.

In the Agile enterprise, implementation of architectural epics is complicated by the fact that the “big-bang-up-front branch-and-merge” waterfall approach is abandoned. Instead, the enterprise commits to implementing architectural epics like all other epics (and features, and so on) incrementally in the main codeline. Doing so means that architectural epics must be split into [Architectural Features](#), which builds the runway needed to host the new [Business Features](#), as Figure 2 illustrates.

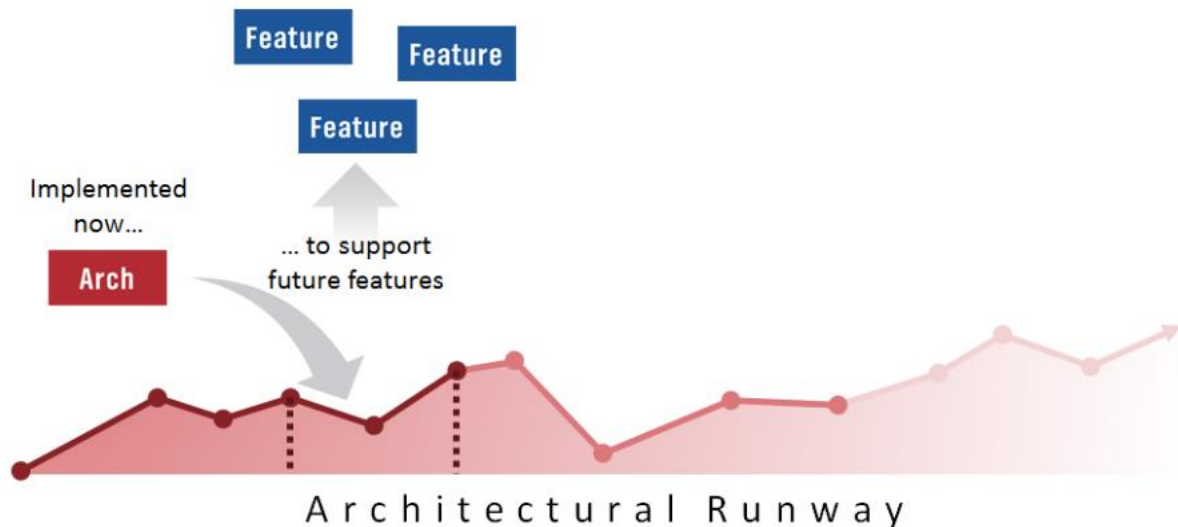


Figure 2. Architectural runway continuously evolves to support future functionality.

Each architectural feature must be completed within a [PSI](#) such that the system always runs, at least at the PSI boundaries. In some cases, this means that new architectural initiatives are implemented piece-meal and may not even be exposed to the users in the PSI in which they are implemented. In this way, architectural runway can be implemented and tested behind the scenes, allowing shipment throughout, and then exposed to users when a sufficient feature capability exists in the next PSI or so.

The concept of runway illustrates how intentional architecture and emergent design effectively complement each other at scale: intentional, high-level ideas in support of future functionality are adapted and instantiated by agile teams; they are empowered to figure out the optimal emergent design.

Principle 3 – Build the simplest architecture that can possibly work

We welcome changing requirements even late in development (Agile Manifesto [2]). Yes, we do, but surely enabling change is facilitated by systems that have understandable designs. As Kent Beck notes: *If simplicity is good, we'll always leave the system with the simplest design that supports its current functionality* [3]. Indeed, at scale design simplicity is not a luxury, but a survival mechanism. There are many considerations that help us accomplish this, to name just a few:

- Using a simple, common language when describing the system
- Keeping the solution model as close to the problem domain as possible
- Continuously refactoring
- Assuring that object / component interfaces clearly express their intent
- Following good old design principles [4,5].

Certain approaches to designing the system, such as Domain-Driven Design [6], usage of design patterns [4], and applying metaphor [3], simplify both the design and the communication between the teams. This “social” aspect of design simplicity is critical as it enables collective code ownership, which in turn fosters feature (rather than component) orientation of agile teams [7]. Most of the approaches mentioned above consider the system as a set of collaborating, meaningful entities. This is the dominant theme in evolving maintainable and extensible solutions which that prevents programs from typical design flaws such as concentrating too much logic at the database layer (often without a real reason) or creating a thick UI, or ending up with humongous, unmanageable classes. That is, simplicity requires design skill and knowledge and agile organizations may effectively leverage Communities of Practice to continuously develop and spread these best practices.

The right balance of intentional architecture and emergent design also allows for generalization of solution elements to address common problems at the component, system or enterprise level.

Principle 4 – When in doubt, code or model it out

Making good design decisions is an art. Often (and thankfully) there are disagreements of opinion amongst team members about which course of action is best. And while Agile teams and programs don’t mind refactoring, they surely want to avoid *unnecessary* refactoring. In order to decide, Agile teams can typically just “code it out”, using [Technical or Functional Spikes](#), and even rapid prototyping. Iterations are short and Spikes are small; the result is fast feedback with objective evidence, which can then be subject to A/B testing by the teams, designers and architects, or even the users.

In cases where design changes are of such scope that spikes and prototyping are insufficient, it is useful to model the problem to gain an understanding of the potential impact prior to implementation. [Domain Modeling](#) and [Use-case Modeling](#) are lightweight Agile modeling constructs that are particularly valuable in these circumstances.

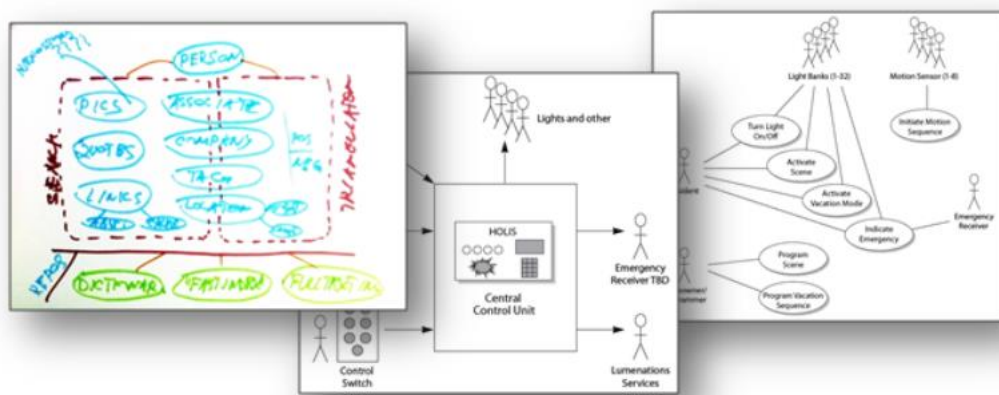


Figure 3. A domain model, system context and use-case diagrams are useful in understanding the system, and in assessing the impact of architectural change.

What’s more, both are helpful; for big decisions, *apply spikes and prototyping in conjunction with modeling* so that everyone can see the bigger picture and the evidence it produces.

Principle 5 – They build it they test it

Testing system architecture involves testing the system’s ability to meet its larger scale functional, operational, performance, and reliability requirements. To do this, teams must typically build an automated testing infrastructure that enables ongoing system-level testing. After all, if it can’t be tested, it is assumed not to work. If the architecture is evolving, testing approaches, testing frameworks and test suites must evolve with it. The responsibility for knowing how to test a system lies with those who collaborate on the design of the system. Therefore System Architects, agile teams and the [System Team](#) actively collaborate in order to continuously [Design for Testability](#).

Principle 6 – There is no monopoly on innovation

In our balanced model, architecture is a collaborative effort of agile teams, architects and stakeholders. This can help foster a *culture of innovation* whereby innovation can come from anyone and anywhere. Many ideas come from individuals, agile teams, system architects, and technical management, but they may also require centralized scaling effort to the program or enterprise level. One of the responsibilities of the Enterprise Architect is to foster an environment where innovative ideas and technology improvements that emerge at the team level do not pass unnoticed, but can be synthesized into the building blocks of the architectural runway. And as Agile can foster the “tyranny of the urgent iteration”, programmatic time for innovation can be built into occasional [HIP](#) (Hardening | Innovation | Planning) sprints.

Principle 7 – Implement architectural flow

Organizations must be able to continuously improve their process of implementing large architectural initiatives that impact the runways of multiple systems. Unless properly managed, these cross-cutting [Architectural Epics](#) will induce excessive work in process and will create long queues in affected Agile Release Trains and can thereby inhibit delivery of business value.

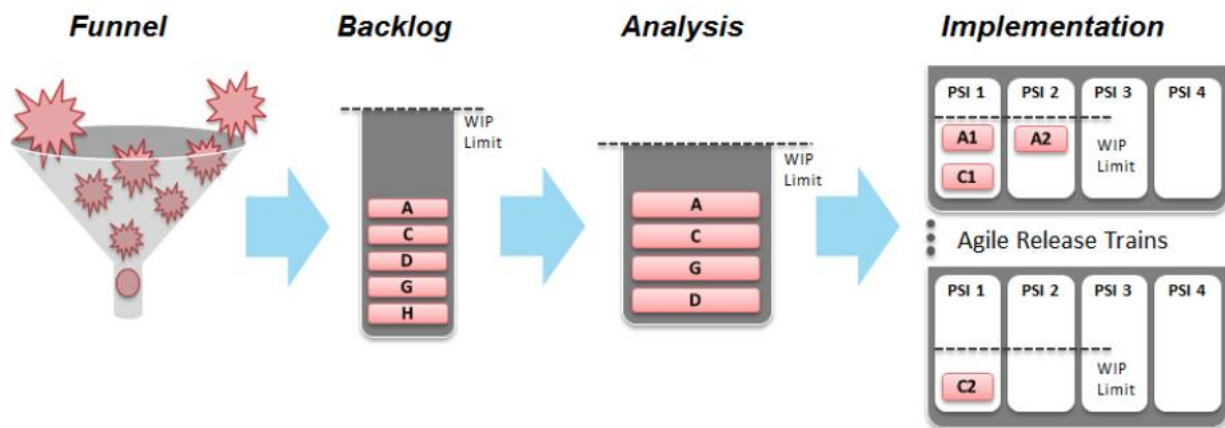


Figure 4. Architectural Epic Kanban facilitates the flow of enterprise level initiatives

To provide for visibility and to help optimize the *flow* of such initiatives, SAFe applies an [Architectural Epic Kanban](#) system, which consists of four states:

- **Funnel**, where all ideas and possible initiatives are collected
- **Backlog**, where the understanding of the epics gets refined
- **Analysis**, where a lightweight business case is developed, and final go / no go decision is made
- **Implementation**, where epics, after being split into architectural features, proceed to implementation by Agile Release Trains

Effective implementation of such a system can be a key factor in achieving enterprise success.

Summary

In this article we introduced the notion of the balance between *intentional architecture* and *emergent design* as a Lean | Agile agile approach to addressing the complexity of building enterprise class software solutions. We noted the power of collaboration as a way to harness the creative power of all participants. We highlighted the need for Architectural Runway— the foundation for future development of business value – which must be continuously extended and validated, implemented incrementally using simple design metaphors to increase maintainability and extensibility. Innovation is an important facet for success, and that comes in large part from both individuals and the collaboration between the roles. Finally, we introduced a Kanban system to enable visibility and flow of architectural initiatives that cut across multiple value streams, thereby helping the organization continuously build an effective technical foundation for arising business opportunities.

Learn More

- [1] **Leffingwell, Dean**. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*, Boston, MA: Addison Wesley, 2011.
- [2] *Manifesto for Agile Software Development*, <http://agilemanifesto.org/>.
- [3] **Beck, Kent**. *Extreme Programming Explained: Embrace Change*, Boston, MA: Addison-Wesley, 2000.
- [4] **Bain, Scott**. *Emergent Design: The Evolutionary Nature of Professional Software Development*, Boston, MA: Addison Wesley, 2008.
- [5] **Shalloway, Alan**, et al. *Essential Skills for the Agile Developer: A Guide to Better Programming and Design*, Boston, MA: Addison Wesley, 2011.
- [6] **Evans, Eric**. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston, MA: Addison Wesley, 2003.
- [7] **Larman, Craig**. *Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum*, Boston, MA: Addison Wesley, 2010.

[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.

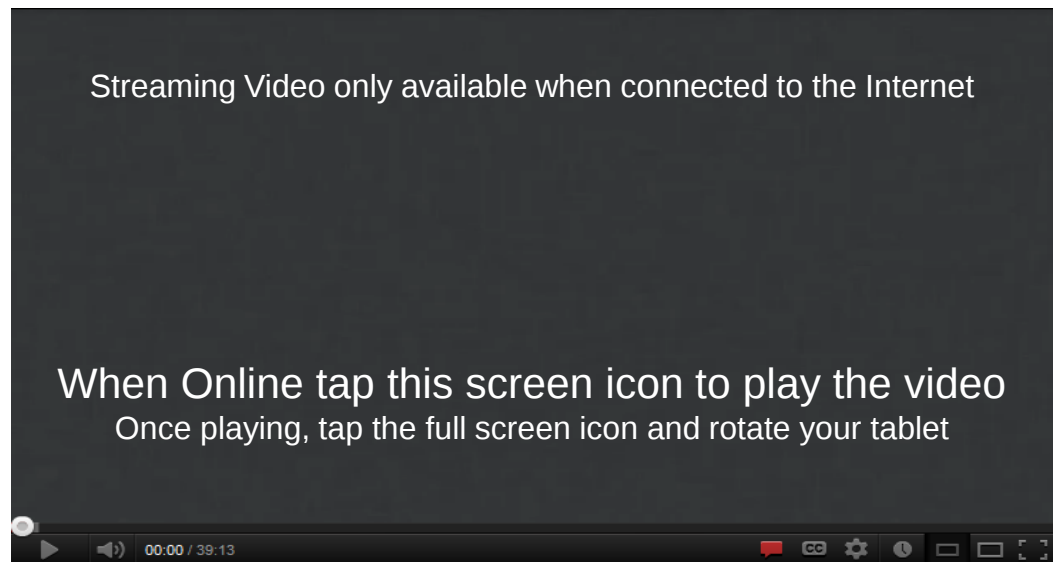
Deliver Early, Often, and Exceptionally with Visual Studio Premium



Cheryl Hammond
ALM Consultant
[Northwest Cadence](#)

Developers using Visual Studio Professional know how the right tools help teams collaborate to deliver great code. With Visual Studio Premium, extend that collaboration to the broader team—stakeholders, testers, and leadership—with all the power of Test Professional and additional Premium-only features to speed the delivery of high-quality, on-target applications.

Expect to see: new features for first-class testing, robust code review workflows, agile management support, and end-to-end insights the entire team can rely on.



Click on the graphic above to play the video



[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](#)

W

e'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

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: *Yes that was the first startup I worked in.*

ALM Mag: Then you co-founded a startup that created Visual SourceSafe?

Brian Harry: *That's right. The startup was originally called Icarus Software and somewhere about a year into it, we got a Cease and Desist letter from a lawyer informing us that they did not appreciate us using their name. Somewhere there was another company named Icarus Software, so we had to do the emergency name change exercise and went through a name search trying to find a name and we landed with the name One Tree Software. That's what we used for the last couple years of the company and then it was ultimately acquired by Microsoft and that's why I ended up here.*

ALM Mag: How did you find that transition from the very small and very entrepreneurial to a still fairly entrepreneurial Microsoft at that time?

Brian Harry: *Microsoft was big_even then, I don't remember how many employees but it was over 10,000, so it was a good size company and it was a big change for me. I never imagined in my wildest dreams that I would ever work at a big company and it was a change from a very start-up focused, everybody does whatever has to be done mentality. We used to have Friday night pizza parties, packing boxes and duplicating discs and you just did what was needed to a role here at Microsoft where I was a developer writing code. You know it's funny, I feel like as in some ways Microsoft is even more entrepreneurial now then it was then, because I think over the years we've more and more realized the importance of having developers engage in the community and really get directly involved with customers. That was something that I missed going from sort of start-up to Microsoft circa 1995 and I think that is something that we've grown into and I'm happy about, having played part in at Microsoft.*

ALM Mag: So the circle has kind of almost become complete?

Brian Harry: *It's not a start-up by any stretch of the imagination but I think some of the aspects of what I loved about start-ups we've managed to create in a little bubble. In fact, even my team and I work out of North Carolina and not in Redmond. I lived in Redmond and worked in Redmond for about 8 years but, I moved to North Carolina back in 2002, I moved back home. We've got a little office there and in fact it was really funny, when I first moved back... my brother who was at One Tree when we got acquired came to Microsoft with me and he moved back at the same time I did and joined my team and it was just the two of us. We set up a little office in an extra room in his house and that was the Microsoft North Carolina office for the first 3 or 4 or 5 months. I would drive over every morning to his house and we would sit there with our little computers on the desk and dream up what eventually became Team Foundation Server. That was kind of the humble beginnings of it and now even though the team has grown, we tried to keep a bit of a start-up feel to it, so it's been fun.*

ALM Mag: Most people associate TFS and ALM with software development, but I'm sure there are probably examples that you have where TFS and ALM is used outside of software development realm. Do any notable examples come to mind?

Brian Harry: *People do use different aspects of it. People use the version control system for document management of all manner of documents, but one of the ones that tends to generate some of the most interesting ones is the web-based sort of project manager stuff. You can use it to manage any list of tasks and prioritize and sort of schedule and assign to people. One of my favorite ones was when we released an on-line service a little over a year ago. We made it public and have gradually opened it up and last fall sort of RTM'd it and made it available to the world. I remember someone telling me that they were using our Scrum board to plan their wedding. They created a project, the wedding project and were using it to plan their wedding. People use it for all kinds of interesting stuff. Obviously they weren't going to use the build system for their wedding, but people use it for all kinds of really interesting things.*

ALM Mag: Obviously TFS has an Agile process template and people use Agile to run their households. Whether it helps them run them more efficiently or not, I don't know... but I have heard of that.

TFS 2012 introduced some really nice refinements to the user experience and one of those areas was around Lab Manager. So, how does TFS 2012 improve upon that Lab Manager experience?

Brian Harry: *Well in 2012 the big thing that we did was a deeper integration with System Center. We introduced Lab Manager in 2010 and the truth is... that product is designed to enable you to much more easily get your software deployed into a test environment and then run your tasks and report the results. At the time we built that product we were kind of ahead of the System Center infrastructure. We used System Center Virtual Machine Manager, but we were building a lot of capabilities on top of it in terms of managing multi-machine environments, connecting to them and deploying software to them, etc. By 2012 System Center had sort of incorporated many of those capabilities, and in fact done, in some ways, better than we had done them, so the big thing that we did in 2012 was to actually remove some of the stuff that we had built in 2010 and rely on, kind of underlying System Center capabilities and through that process we were able to simplify it and make it better.*

ALM Mag: Interesting, it's kind of reverting back to... don't reinvent the wheel, especially if the wheel somewhere else handles something a little bit better and you can leverage it.

Brian Harry: *Yeah, that's right.*

ALM Mag: I response to a survey, readers of the magazine, wrote in some questions that they would like answered and so some of these questions come from that survey.

: One of the questions is: When will a TFS team make moving between process templates easier, i.e. a completely automated move? So what I'm hearing into this is that they want to move from one process template possibly in one version to another process template.

Brian Harry: *I think it's a good question. It's one that we receive a fair amount of feedback on over the years. Let me break that down into two scenarios though. One is upgrades. I have TFS, I have a process template and I want to upgrade. Then there's another scenario in which I have TFS, I've got one process template and I want to move to a different process template, an arbitrarily different process template. Those two are different, they are related but they are two different problems. Let's talk about upgrade first... and there are even two pieces of that that I want to talk about. We did some work in 2012, so the first thing is let's say I'm using 2010 I've chosen the actual template and I want to upgrade to TFS 2012. Well first of all it just works. You upgrade to 2012 and it just works, but you are still using the process template from 2010. You think well gee, you added a bunch of features, you added Taskboards and you added Code Review and you added these cool features, I sure would like to be able to use those but they require capabilities in my process template that aren't there. Well we built a feature in 2012 that we call Feature Enablement that allows you to sort of go through and say I want to add Code Review, I want to add Taskboard support and it will go out and add the necessary capabilities to your current process template to enable that. Let's call that a baby-step in the direction of where we want to go. The second step is, hey you know what I really want is, I was using Agile 4.2 and I want to go to Agile 5., I just want to move the whole template forward. I'd say that's a fairly high priority for us, particularly in the context of a service, because right now we don't allow customization of our Service Process Templates, exactly because we are trying to make the upgrade process very reliable on the Service. So, that's a problem that we are going to tackle fairly soon, in terms of enabling really full transformation from a previous version of a template to the next version of a template. Then there's the general, I have arbitrary template I want to move to another arbitrary template, that's a really hard problem. It is a general schema transformation problem. I have one schema and I want to transfer it into another schema and when you do that, there may be no good correlation between those two things. And if you look at the SQL world, how does that work? You get these complex transformation pipelines or you plug in modules that maps schemas and we don't want to try to create anything that complicated. So what are we going to do there? I don't know. We'll see... we are not going to do anything soon there, but we know that this whole space is an area that people care about.*

ALM Mag: There was another question about when will customization of TFS Service be available?

Brian Harry: *That's something that is important to us to get to. I don't think it's going to be in the first half of this year, but I hope it will be in the second half of this year. You know... obviously, any peering into the future is a dangerous thing. Because you never know. We are an agile team, we practice sort of as an Agile practices and therefore, I don't know much concretely beyond the next few months. I can speak fairly concretely about that... beyond that we have a prioritized backlog and we kind of roughly know what order we want to do things in but I don't have timeframes.*

ALM Mag: Yes in true Agile fashion or Agile spirit. So for co-located teams, how is TFS better than cards on a wall?

Brian Harry: *How is TFS better than cards on a wall for co-located team, certainly for distributed teams it's a big deal, it makes a big difference. I guess I'd say the advantage is you can access it from wherever you are, even for a co-located team... people work from home, you have to get up from your desk and walk to the board and make changes. Isn't it just easier to be able to associate the task with the check-in, do your check-in and it automatically advances to the next state on the board. We can certainly remove a little bit of that friction in the process. Ultimately, I think we'll have other advantages in terms of what you can do virtually, with the ability to add a new column to your taskboard or where you are located let's redraw the taskboard, re-affix all the "stickies" up there. Electronically for those kind of things you just click a button, there's a new thing, drag some new cards into it, it's easier, certain things are easier to do virtually than physically.*

ALM Mag: Yes, I've used both and I've used both together and I think that having the taskboard available to even with co-located teams, including people that work from home, and also being available to management and stakeholders who are not necessarily in that same area adds value. On the flipside of that coin, having a physical board, I like the idea of having people actually getting up and physically moving, getting out of their seats from time to time. Although the new taskboards, the electronic taskboards where you can use Surface and Touch and move items around are pretty nice.

Regarding the taskboard, there is another question that came in about some folks want more flexibility in the work item types that it can be used for. Do you have a comment about work item types in this context?

Brian Harry: *I think so, I'm going to have to guess at exactly what is meant because you can put any Work Item Types you want on the taskboard and you define them. However one of the things I'd say is... here's a common issue. The taskboard is set up by default to share your user stories on the left, tasks in the board and you advance tasks through the board. There are kind of two scenarios where I see people struggling with that. One is Bugs. So Bugs are not a user story and they aren't exactly a task. You could treat it like a task but they are not associated with a User Story so where do they go on the board? Right now the answer is they don't. You don't put the Bugs on the board, so you don't advance through them. You have to treat Bugs differently. That's something we would like to address. We would like to be able to have you have your Bugs on the board and treat them just like a Task without the overhead of having to require some goofy User Story that you associate all your Bugs with. And so there is a flexibility in terms of how the board can work when you don't have kind of a strict hard rule relationship between some container type of Task or Requirement or User Story, or whatever it is and a child type Task or whatever. The other one that I sometimes hear that people would like more flexibility around are multiple levels of hierarchy. When I have Epics, Epics are*

decomposed to User Stories and User Stories to Tasks. Our Taskboard right now is optimized for a two-level hierarchy, User Stories and Tasks, Requirements and Tasks, whatever you want them to be. It would be nice if there were kind of a way to just drill through multiple hierarchies so that on a larger team where I've got larger more complicated work information it would be better. Those are the two areas that I mostly hear feedback around.

ALM Mag: OK, a question came in regarding Kanban customization, if there's a timeframe in mind to be able to do customization for Kanban and will it be available on both the Team Explorer version as well as TFS Service versions?

Brian Harry: *Thank you for the softball question... I love that one. So I just announced yesterday, in my ALM Summit Talk, that we're adding Kanban customization. It will show up on the Service, I think the week after next and it will be included in Team Foundation Server Update 2, so you'll get it on premises when Update 2 ships. It will be in the next community technology preview of Team Foundation Server and it will be available on the public service in the next couple weeks.*

ALM Mag: Good, so along that same line of softball questions and recent announcements, there were some announcements around DBCS and GIT. There are several questions that came up regarding that even though the announcement was very recent. From a high-level perspective initially, what can you tell us about the whole DBCS focus?

Brian Harry: *So DBCS has been a growing phenomenon it has gathered a tremendous amount of momentum. It's had its roots, in kind of the Open Source community and in worlds where you had very distributed teams, no organization, in terms of a corporate organization around them, really peers collaborating together and this notion of sort of a wild west of exchanging ideas and changes. That's kind of where it's originated, but over time, it enabled a set of workflows even for teams that are more structured. It enables a great offline experience, it enables some great experimentation capabilities, it enables some great flows, innovations around the pull request workflow, that enable this change in control. The guy authoring the software can make suggestions but the person who owns the software can decide whether to accept those changes or not. This is an interesting workflow in teams in the same organization. It's just a set of workflows that by no means is right for every organization and every team but increasingly as software architecture is changing to more loosely coupled systems, more componentized, more built by distributed teams even within the same organization, that these kinds of workflows are becoming more popular. It's something we've been watching for a couple years now. We've gotten a lot of requests on our user board site to be able to provide DBCS. In fact it was the number 3 request on user boards, I think it had 1380 votes or something. A lot of people wanted a DBCS workflow in TFS. So after we shipped TFS 2012, in fact even as we were starting to ramp down on TFS 2012 we picked that as one of the big next investments, to be able to provide that workflow for TFS customers.*

ALM Mag: Is this capability available in both the Team Explorer version and TFS Server?

Brian Harry: Yes, it's available today on TFS Service and we've announced that it will be available in the next on-premises version of TFS. We sort of called that TFS 12, the last version was TFS 11... it gets really confusing, because TFS 11 shipped in 2012 and was ultimately called TFS 2012 and TFS 12 will ship after because it's 2013... it will be sometime after. We haven't announced any date and I couldn't even say exactly when it would ship but it'll be in the next version of TFS.

ALM Mag: Ok, so I think yesterday in your ALM Summit talk you announced GIT support for Visual Studio and TFS, how does this GIT support help Agile Scrum teams?

Brian Harry: Well I'm not sure how to break down that question, so I can answer sort of generically, how does DBCS help Agile and Scrum teams as maybe the first part of that question and fundamentally that's just about enabling a more flexible workflow. It's easier to manage Branches, it's easier to move code flexibly between different Release Branches which can enable teams that work in certain ways to be more agile and more flexible so that there's some advantages to DBCS for certain work styles. With respect to GIT specifically, why is that kind of a good thing? Well I think there's a couple reasons. One it's really popular and it's got a very established eco system out there, a set of tools around it, a set of practices. There's a community for sharing "best practices" and so there's an advantage and why I think it's a good choice as it helps not just agile teams but certainly anybody adopting a DBCS solution. And secondly, again another part of community is, almost anybody out there building an application is looking at how they can build upon the body of work in the industry. Why reinvent the wheel if you don't have to. A big body of that work is in the Open Source community and being able to figure out how to consume open source, to use open source, to advance whatever innovation you are trying to bring forward is a value for any team. I want to focus on new innovations not on reinventing something somebody's already built. So whether its, I want to build on Hadoop, or I want to build on... whatever, let me take advantage of that work. GIT obviously is a big player in that space and by providing integration with GIT we make it even easier for people to participate in that open source community both to consume and to contribute back, we think these are good advantages of GIT.

ALM Mag: Are there specific ALM benefits gained by using GIT, do you gain or lose any benefits in the ALM practice?

Brian Harry: Ah a very good question. I wouldn't have said that there are any inherent advantages to GIT with respect to ALM. GIT is a version control system. ALM is much much broader than that but what I certainly will say is, our approach to GIT, is to integrate GIT into our ALM workflow, so it's not just... hey we're going to bolt this GIT thing on the side of any need we have. We have a GIT version control system, but rather the way we are thinking about it is... hey we have two workflows, there is a centralized workflow and there's a distributed workflow. GIT happens to be the kind of approach we've taken to be our distributed workflow, but both workflows have full integration into the ALM lifecycle whether it's the way you do your

planning, or the way you do Builds or any Continuous Integration, Gated Check-ins, all those kind of capabilities, work item linking and the way you're sort of managing your work. My Work, Code Review, Testing, Deployment, all of those pieces that are part of the overall application life cycle we will be integrating GIT with. You will feel like it's a native part of Team Foundation Server and Service, so yes absolutely we think that's an important part of it.

ALM Mag: And as I was watching your demonstration yesterday it really seemed like a very seam-less and tight integration.

Brian Harry: *We hope so, there's a lot of work left to do, but we feel really good about the initial steps that we've made and it's something we're going to iterate on rapidly. We're on a 3-week sprint cadence and we ship something every 3 weeks to the Service. We've also been shipping some kind of Community Technology Preview of even some of our on-premises bits every 3 weeks now via the VS plugin. Whether we're going to continue to ship that literally every three weeks or maybe we do every other sprint we haven't decided yet but we'll continue to update it regularly and will continue to make progress.*

ALM Mag: So how should teams decide whether to use Team Foundation Server Version Control or GIT? TFVC or GIT?

Brian Harry: *Yeah, so that's a tough question. We have tried to take some cracks at answering that in fact Matt Mitrik wrote a blog post on our ALM blog just recently with a section on how to make that decision. To some degree I'll say, it's like which is better apple pie or cherry pie, they're both pretty darn good and your tastes will have some impact on which one you think is better. It's going to be driven by your work style, your preferences, but both are very capable and work for teams. Now I'd say that Team Foundation Server Version Control is generally going to scale better. If you have a very large team, very large code base, and perhaps more centralized control... it's going to scale better. We have in Team Foundation Server Version Control, projects with millions of files in them and where people are having to work on hundreds of thousands or millions of files at a time to do large scale refactorings and changes, Team Foundation Server Version Control works really well for that, for very small teams, distributed, loosely coupled with I'd say non uniform levels of contributorship. Example, I've got somebody who is a clear owner and other people who are more reviewers and maybe occasional contributors, DBCS is a really great solution because the sort of flexibility of creating Branches, of creating forks, of contributing sort of changes for review that is a much simpler workflow in the DBCS system than it is in the centralized version control system. It's not that you can't do it in a centralized version but it's hard.*

ALM Mag: So we've mentioned the CTP platform or process several times. Is the rationale behind placing these new capabilities out there on TFS Service first because it's a more controlled environment, and then rolling them out to the Team Explorer environment?

Brian Harry: *Yeah, that's right... it's just easier for us quite honestly. We can be more agile on the Service as shown by us shipping on the Service every three weeks. If we ship something on-premises it's going to get installed on thousands, hundreds of thousands or millions of systems depending upon what it is and how long it's out there. Our ability to update that is super hard. It's hard to get to those people to find the time to install the systems., The number of configurations it gets installed on is crazy large and that means our test matrix becomes crazy large. One of the nice things about the Cloud is that there's exactly one config. that we have to install it on, so we can focus the testing on that. We have complete control over deployment so we deploy exactly to one system. Generally deployment for us takes a few hours from beginning to end, so we have a new Build... a few hours later, it's on the system. We can just be so much more agile that way. It's by far the easiest way for us to preview early features, get feedback, work on stability, make sure that the stuff is ready. Then once we get to the point that, hey we feel like we got a winner, we have a feature or capability that people are really going to like, then we can go through the rest of the effort of the full test matrix, validated in all the configurations and ship it to an on-premises customer and let them install it in their own Data Center*

ALM Mag: OK, so one of the questions came in, and again regarding TFS Service, how is TFS Service different from an on-premise installation of TFS?

Brian Harry: *Yes, so there's a long list of differences. Let me start by saying, in principal it's the same, it's the same code base. It's one code base that runs on-premises and in the cloud. There's obviously some factorings where there are differences. Let's take an example, the identify system in Team Foundation Server is pluggable and for on-premises there's an active directory based identity system, for the Cloud there's a Live ID, called, Azure Active Control Services plugin that uses Live ID to do authentication, but 90% of the code is the same. We sort of factor it to allow modules to customize to the environment. Now that's one way to think about it, another way to think about it is, there's a set of features which we've not yet done that factoring and brought them to the Cloud yet. Things like reporting, is an area where we have not yet brought it to the Cloud yet, Sharepoint integration, Project Server integration, System Center integration. We have not yet brought those capabilities to the Cloud, we will, we just haven't yet.*

Then there are certain restrictions that we put in place to make our job easier for now and the lack of process customization that I talked about earlier was a good example of that. There's nothing that prevents us from enabling process customization other than as soon as we do, upgrading the Service becomes more complicated because we've got to be able to deal with how we get each team's process updated and when that's right for them cause maybe I'm in the middle of doing a Release of my software and I don't want you messing with my process right now. I need it to stay fine, but yet I've got to upgrade the server. We've got to work through all that stuff, so yes there' are a fair number of differences but there's an awful lot of capability up on the Service now. In general the on-premises product has more capability, particularly around enterprise grade capability, such as reporting, Project Server and Sharepoint.

ALM Mag: That brings us to probably the number 1 question that people ask, something that has, I guess proven to be difficult but, when will TFS allow project names to be changed?

Brian Harry: *Oh my gosh, yeah... you know that is one of my great humiliations to the fact that you can't rename projects in TFS. That problem, I know is so hard for people to understand, why don't you just fix it, and it seems like it ought to be so easy. We made some design decisions way back in 2004, maybe even in 2003, it was a long time ago, that just turn out to make it much harder than you would possibly imagine. It's not impossible, it is doable, but it is much harder than you can imagine. We've costed it before, we've looked at it, it's months and months worth of work to be able to do that. We really think that we need to go back and figure out how to go back and rename that project. We need to revisit some of the design decisions that we made, change the middle way that will ultimately make doing the rename easier. We've now done some design work on that., We know what changes, what design decisions that we want to go back and revisit and what changes we want to make. That work is now on the backlog, . We had a long debate about, do we want to do that work or do we want to do the GIT work first. We ultimately decided to do the GIT work first and do what we call project refactoring second. So once we get the GIT stuff over this surge of effort, that's probably going to be the next thing that we work on. So it's you know, I feel terrible it's taken so long, we will get there.*

ALM Mag: Hmm, that's probably good news for a lot of folks. It seems to be a hot topic.

So, what other capabilities can you talk about that are coming in Visual Studio 2012, Update 2 and beyond?

Brian Harry: *Yes, so ahh let's see. I demonstrated yesterday the Web Test Case management which I'm very happy about it. So let me start with a scenario. The scenario, the sort of most common one is, we get customers who use our custom tools and they like them, they plan to test, they create Manual Test Cases, they do Exploratory Testing, but they need to run sort of final validation of their software and that final validation needs to be done in a production sort of... how do I say this, an environment that exactly matches the production environment. They're not allowed to install anything on that environment that's not going to be in the production environment because it could somehow affect the results. The problem that we have today with our Test Professional product is that your manual testing has to install a client on the test environment to be able to run through your test, and this has been an issue for customers., So what we've done is we added manual testing experience to our web UI so that now you can run your manual test strictly from within a web browser without installing anything on the system. It doesn't even download anything into the browser cache other than HTML and Java script. There's just a very simple impact to the system, so that's a feature that we've done that we're pretty happy with. There's the Kanban customization that I talked about and we're excited about, there's Work Item Tagging which I also demonstrated yesterday which is basically a way, and this is going to alleviate a little bit, and I do mean only a little bit, but a little bit of the lack of Service customization. Now I can*

categorize and flag and filter work items with custom values without having to customize the process template., So we added this notion of Tags. I can just tag work items with any arbitrary tag, Hot, Blocked, Front-end, Web, whatever I want, and then I can use those to filter my results and look at various sets of Work Items. This is one of the main things we do when we're customizing things, is to find ways to pivot and look at Work Items. So that's coming... it's on the Service today and it'll be in Update 2 for the on-premises product. You know if I think a bit more broadly in Visual Studio, we've done a ton of stuff, a bunch of unit testing stuff, we did phone unit testing support, some improvability to manage sets of unit tasks... that we call Playlists to create lists of unit tests, more ways to filter and pivot unit tests, and we brought back the Blend features. We had them in VS 2010 but in all our work to get the store application stuff done, we were not able to get everything done for VS 2012. So Blend is now back in Update 2 with Sketch Flow support, WPF and Silverlight support, so that's cool... we're very happy about that. I have to scratch my head a bit... it's a long list and then there's you know I did a big blog post with a bunch of screen shots and stuff yesterday and then there's the list of stuff that we haven't even talked about yet. So there's probably another 20 or 30% that didn't make it into this early CTP yet and will show up in the next or following CTP. So, a lot of good stuff... Update 2, it's funny some one did some math yesterday I saw on Update 1 vs Update 2 and I think Update 1 had like 45 features and someone done some math on Update 2 and found that it had 46. Update 2 is about the same size as Update 1, a teeny bit bigger but there's a lot of value in both of them.

ALM Mag: Maintaining a good cadence, a consistent cadence. So from a higher level stand point, what do you see as the direction that ALM and TFS are heading in?

Brian Harry: We think the... kind of the big industry challenge right now is dealing with the rate of change in everything, whether it's the rate change of technology, the rate of change of business model, the rate of change of consumer behavior. All of these things now change so quickly. Fads come and go and you hear people talking about windows of opportunity and if you don't or you're not able to respond in that window of opportunity and you're not able to track the changing preferences at the rate they're changing, you're just left behind and you lose. So the competitor that's able to keep up with the rate of change is the competitor that wins. Everybody's trying to figure out how to be more responsive and when we think about that from a software standpoint we think about that as iteration. How do I go from idea to implementation to deployment, usage, learning from that usage, to the next idea and how do I turn that cycle as fast as I possibly can? I've got to be able to do that faster than my competitor and so what we're trying to do is build a set of tools that help with that. Now of course that's a much bigger problem than just tools. There are a lot of process and skills problems as well. But we think with tools, we can enable, we can help a lot, we can make it more efficient so that's kind of the background of the problem. More concretely when we think about the next year or two where the larger portion of our investment is going to be, it's going to be in the, how do I get the software from the developer's desktop? The developer's now written something and they're happy with it and they like it. How do I get that in the users hands, learn... see how they use it and what the outcomes from their use of it are, and turn

it into “learnings” for the next idea? So kind of deployment, monitoring, usage... you know understanding, learning, and back into idea. That’s kind of the phase that we think is the most underserved for developers today and people have the least good practices and maturity in that space.

ALM Mag: Hmm, terrific. Yes things are definitely moving at a faster pace and providing the tools and the processes that can help people stay abreast of those changes I think is a key direction to be heading in. That’s great!

So today we’ve been talking with Brian Harry, Technical Fellow and the leader of the TFS platform development here at Microsoft and we want to thank you Brian for taking this time to share your insights and thoughts with the ALM Mag community.

Brian Harry: Thank you... I enjoyed it very much.

[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

These exams can be taken in any order but we'll walk through them in numeric order over this three part series.

Exam 70-496: Administering Visual Studio Team Foundation Server 2012

On September 25th 2012 Microsoft released the Administering Visual Studio Team Foundation Server 2012 exam. This is the third release of an exam focusing on administration and configuration of TFS with the two previous exams focusing on TFS 2010 (exam 70-512) and TFS 2005 (exam 70-510).

Microsoft Learning provide a dedicated page on their website for each exam and you can find details of this particular exam at <http://examcr.am/TyAYs3>

Whilst the specific audience for this exam is described on the exam page, I would summarise this from experience by saying that this exam would be well suited for the “TFS Guru” or “TFS goto person” within your organisation. Remember this exam is not for people that just use TFS to support their development, but rather it suits the person that installs, configures, customises and maintains the health of your TFS server infrastructure.

Exam Details

Exam Code:	70-496
Exam Title:	Administering Visual Studio Team Foundation Server 2012
Released:	September 25 th , 2012
Pass Mark:	700 (70%)
Questions:	Approximately 45 questions
Time:	Approximately 145 minutes

Exam Preparation Best Bet

If you’ve got limited time or you’re feeling pretty confident, I’d suggest you focus on the TFS 2012 Installation and Administration guides. These are both freely downloadable CHM files, available on the Microsoft website at <http://examcr.am/10Ln5E4>

Download both CHM files and read them to ensure you understand the majority of the information covered. Look for knowledge gaps and focus in on those areas for your study. For example, you may have a single combined application tier/data tier configuration in your office. You will still need to know how to install and configure a TFS infrastructure comprising load balanced multi-server TFS configurations.

Exam Preparation – a closer look

The 70-496 exam comprised four key topic areas. These areas are well described on the Microsoft Learning exam page (<http://examcr.am/TyAYs3>) but let’s take a look at how we can focus our study using the free, online MSDN library.

Install and Configure Team Foundation Server (25%)

This topic covers the following high level areas.

- Install Team Foundation Server
- Configure the application tier
- Migrate and upgrade Team Foundation Server
- Install and configure Lab Management
- Install and configure multiple build agents and controllers

Much of the information for this topic can be found on the MSDN Library under the topic entitled “Installing Team Foundation Server and Visual Studio ALM”. Navigate to this topic at <http://examcr.am/XFZmU5> and read all child nodes underneath this topic on MSDN.

Manage Team Foundation Server (26%)

This topic covers the following high level areas.

- Manage Team Project Collections
- Configure for backup and recovery
- Monitor server health and performance
- Operate Lab Management
- Manage security
- Manage reporting for Team Foundation Server

The MSDN Library link for this exam topic is titled “Administering Team Foundation Server” and can be found at <http://examcr.am/SNTBGT>. Remember to read all child nodes underneath this topic on MSDN.

Customize Team Foundation Server for Team Use (24%)

This topic covers the following high level areas.

- Configure a team build definition
- Manage build execution
- Modify a process template
- Configure a team project
- Apply Work Item customizations
- Create Work Item queries
- Configure client connectivity to Team Foundation Server

There are a couple of MSDN Library links for this exam topic. Try looking at “Define Your Build Process” (<http://examcr.am/UqXaIR>) and “Customize Process Templates” (<http://examcr.am/14eEJVm>)

Administer Version Control (25%)

This topic covers the following high level areas.

- Create and manage workspaces
- Configure version control proxy
- Configure team project version control settings

The fourth and final exam topic area also has two primary locations on MSDN for you to focus your study. Firstly look “Using Version Control” (<http://examcr.am/VcESC9>) and then “Team Foundation Version Control Command Reference” (<http://examcr.am/W5ezkY>)

Exam Preparation – Online Webinars

My colleague, Nick Hauenstein and I, are running a series of free online exam preparation webinars to help people get ready for the exam. You can find out more about these and register online at <http://www.quicklearn.com/exam-cram>

Good all-purpose resources

The following two books represent great resources, not just for the exams but for anyone working with or supporting Microsoft Team Foundation Server.

1. Professional Team Foundation Server 2012

Authors: Ed Blankenship, Martin Woodward, Grant Holliday and Brian Keller.

You can find it on Amazon at <http://examcr.am/tfs-2012>

2. Professional Application Lifecycle Management with Visual Studio 2012

Authors: Mickey Gousset, Brian Keller and Martin Woodward.

You can find it on Amazon at <http://examcr.am/alm-2012>

That concludes part 1 of this series. See you next month for the second instalment which focuses on exam 70-487: Software Testing with Visual Studio 2012.

[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



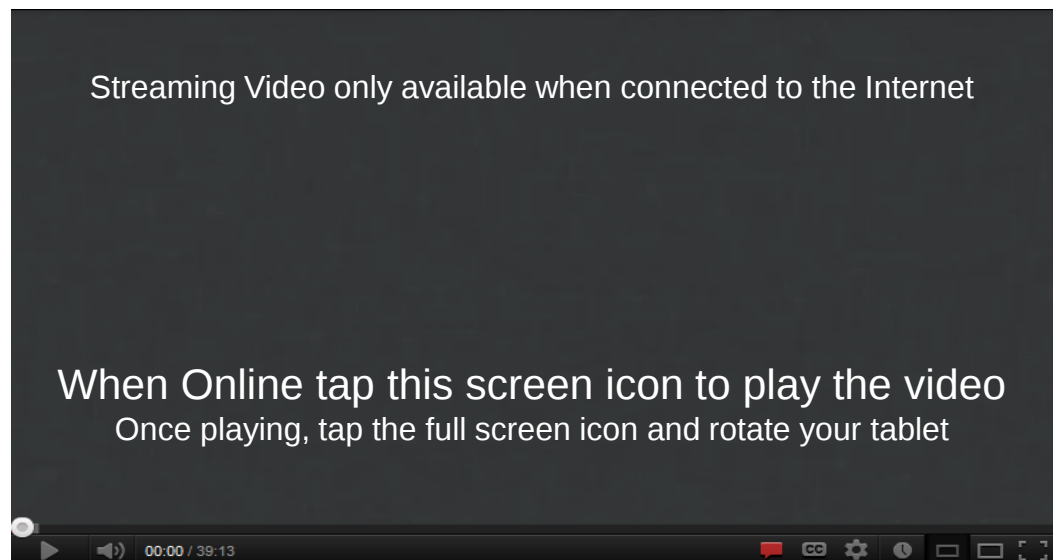
Mohamed Radwan
Senior ALM Consultant
[Marvel ALM](#)

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

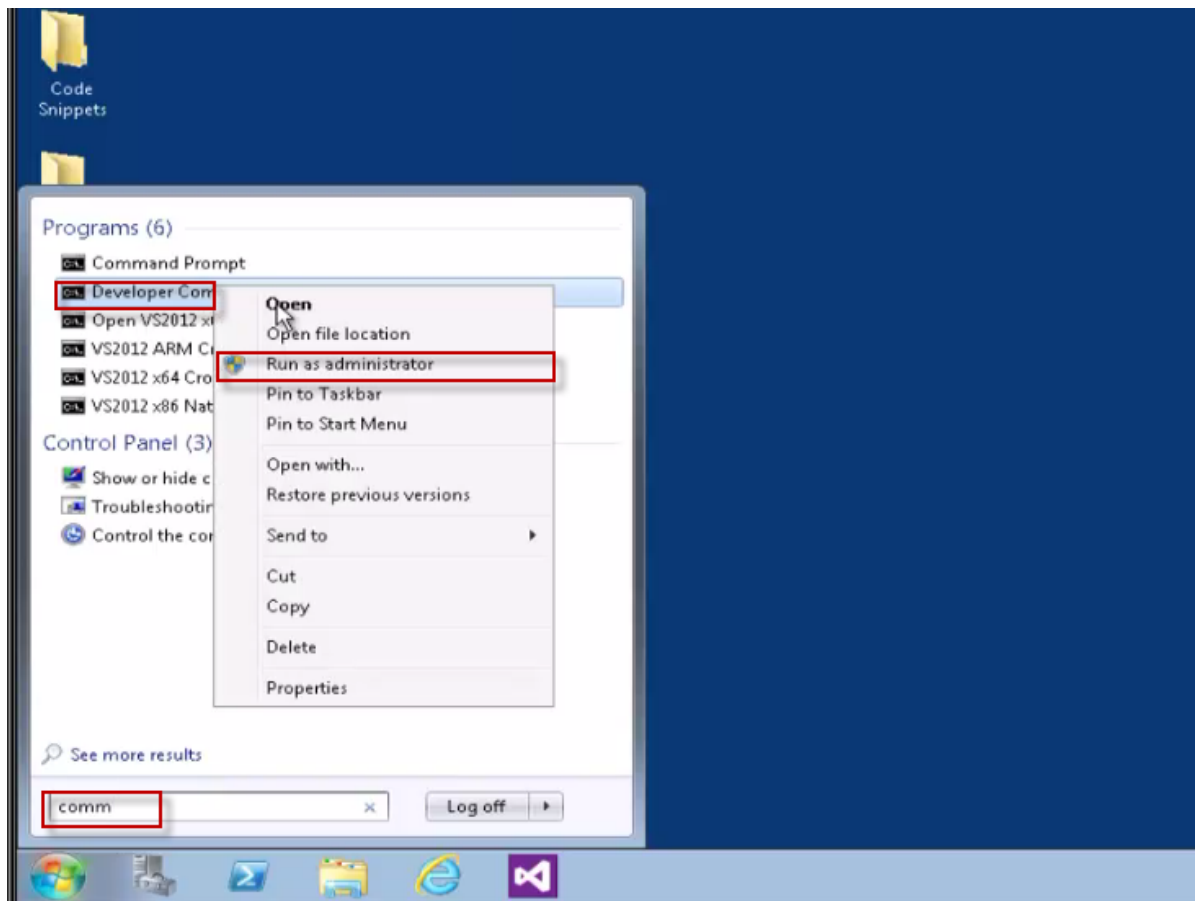
Unable to determine the workspace. You may be able to correct this by running 'tf workspaces collection:TeamProjectCollectionUrl'.

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

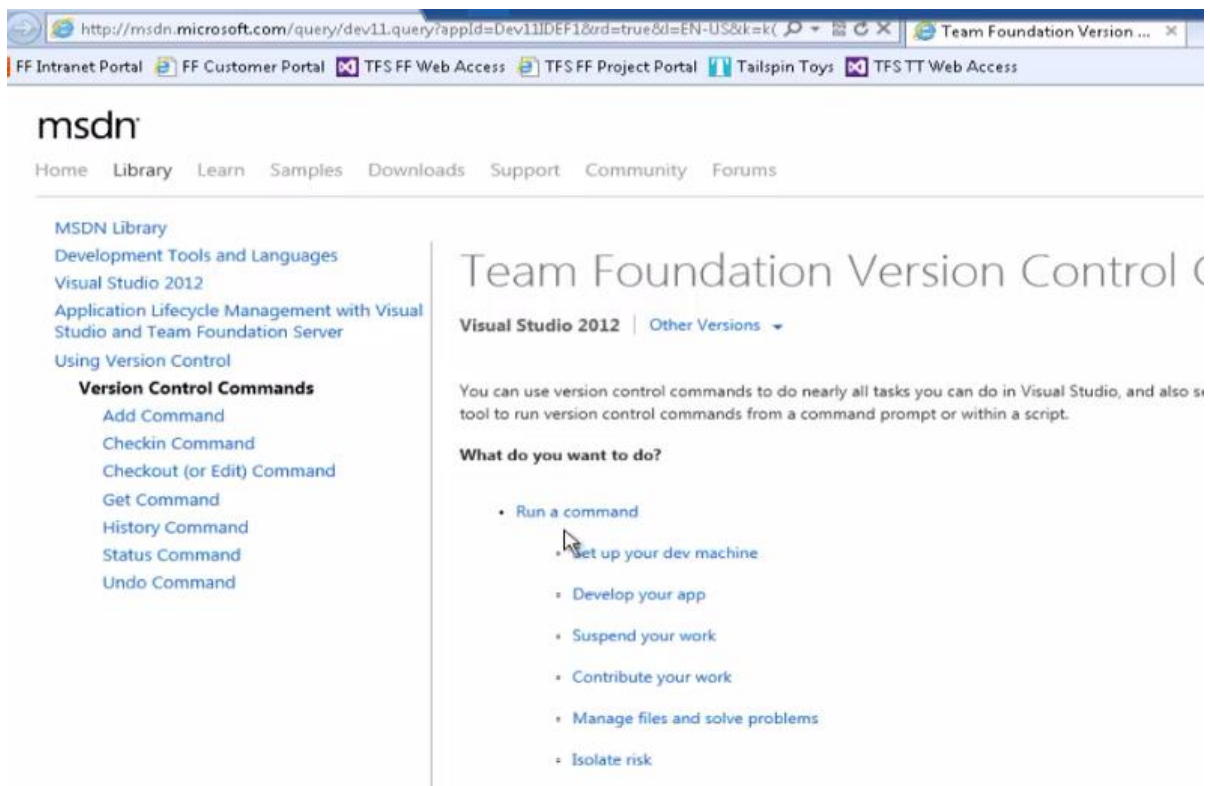


Click on the graphic above to play the video

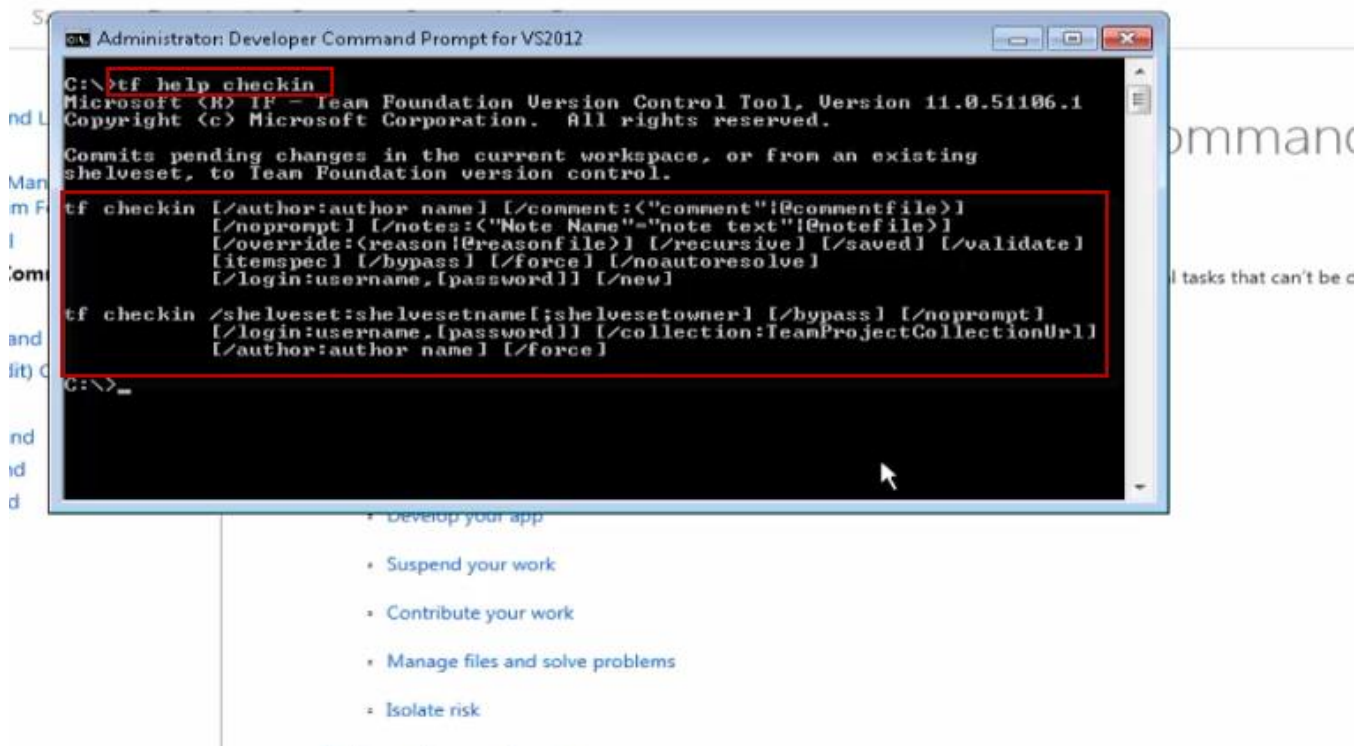
Open the Visual Studio Command Prompt as Administrator, type ***tf msdn***



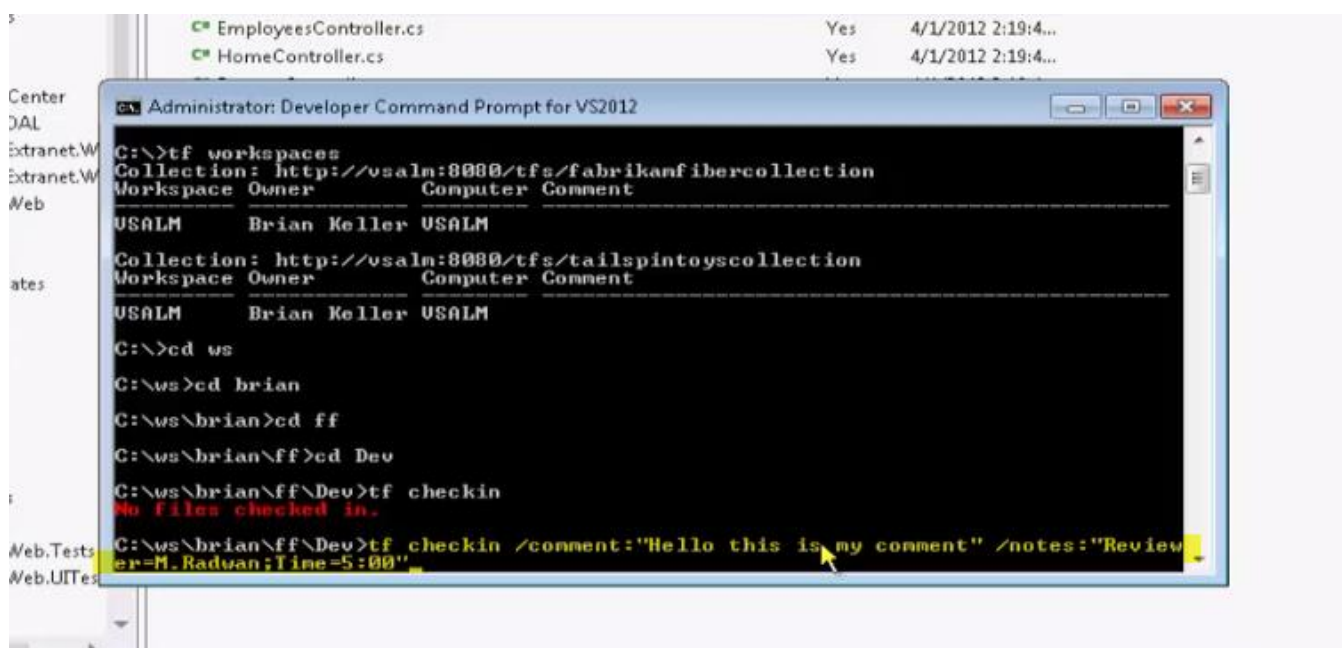
this will display the help



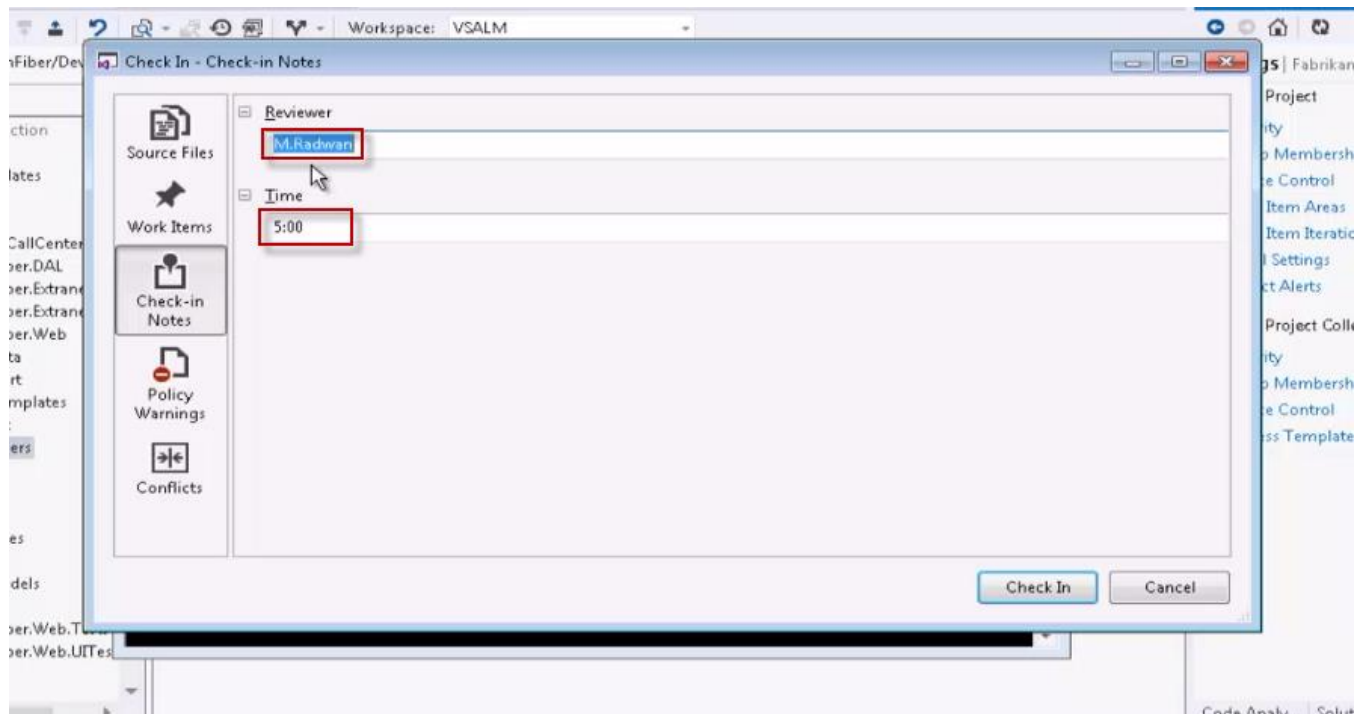
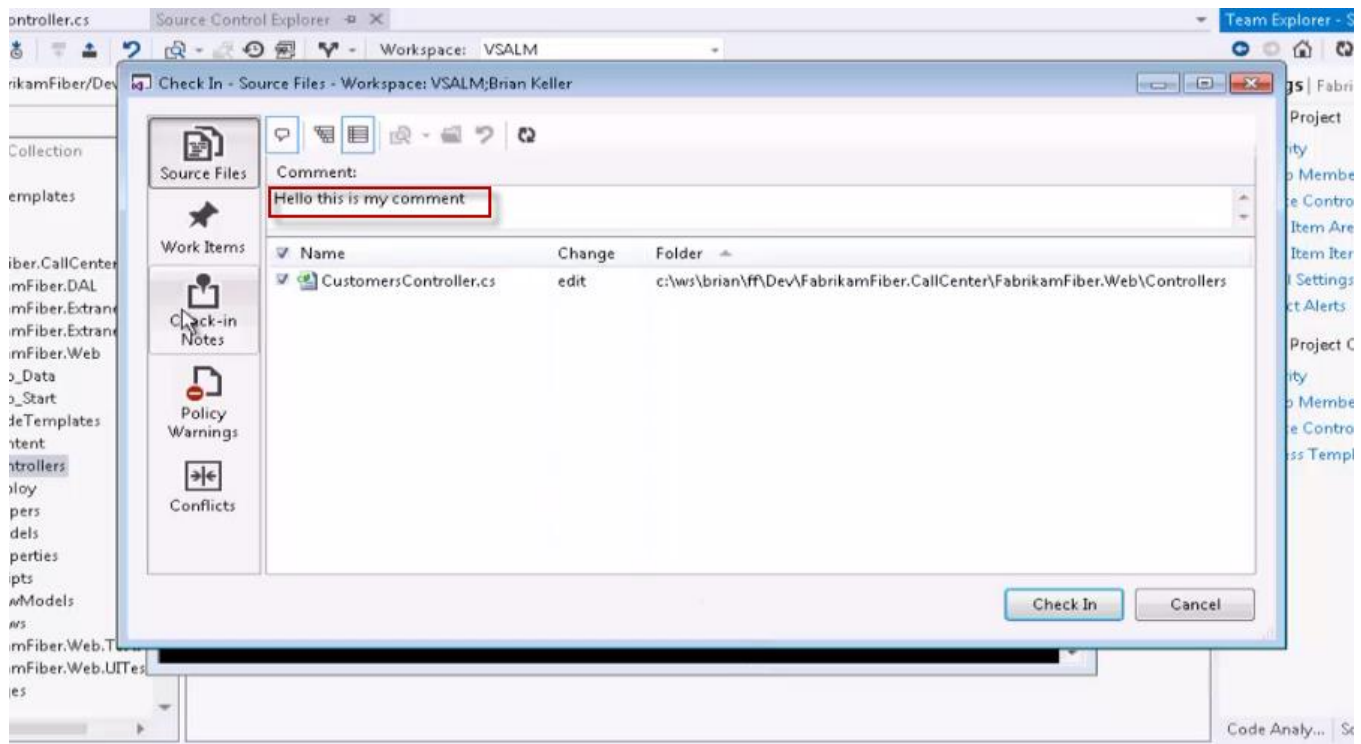
Type ***tf help checkin***, to display the help of the checkin option



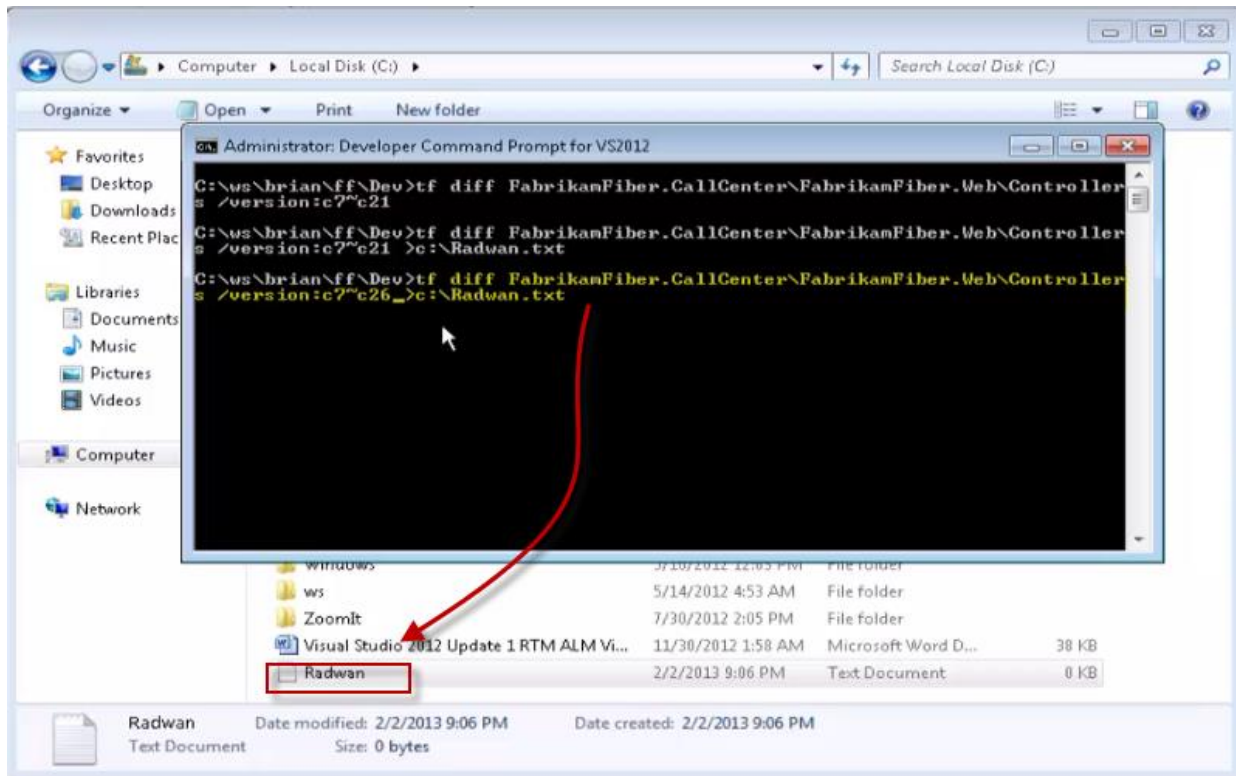
We will go to the project settings and add 2 notes, Reviewer and Time, after that we will get back to the command line and type ***tf checkin /comment:"Hello this my comment" /notes:"Reviewer=M.Radwan;Time=5:00"***



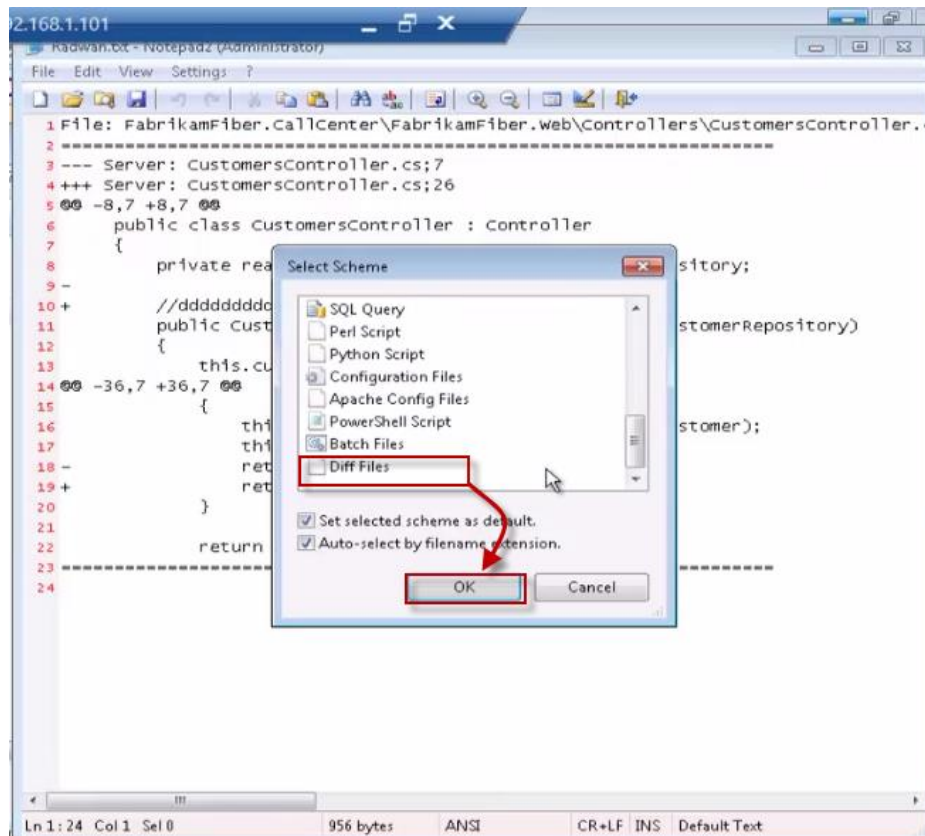
This will display the check-in dialog-box with the typed comments and notes.



Now we will output the differences between 2 changesets in a file, type
tf diff FabrikamFiber.CallCenter\FabrikamFiber.web\controllers\ /version: c7~c26 >c:\Radwan.txt



Open the file using [Notepad2](#) and select the **Diff Files** schema (Into ALM with TFS)



```

1 File: FabrikamFiber.CallCenter\FabrikamFiber.web\Controllers\CustomersController.cs
2 -----
3 --- Server: CustomersController.cs;7
4 +++ Server: CustomersController.cs;26
5 @@ -8,7 +8,7 @@
6 public class CustomersController : Controller
7 {
8     private readonly ICustomerRepository customerRepository;
9     //dddddddddddddddddddd
10    public CustomersController(ICustomerRepository customerRepository)
11    {
12        this.customerRepository = customerRepository;
13    }
14    @@ -36,7 +36,7 @@
15    {
16        this.customerRepository.InsertOrUpdate(customer);
17        this.customerRepository.Save();
18        return RedirectToAction("Index");
19    +    return RedirectToAction("Indexaaa");
20    }
21
22    return this.view();
23 -----
24

```

To view a file a specific version type
tf view FabrikamFiber.CallCenter\fabrikamfiber.web\controllers\CustomerControllers.cs /version:c26

```
y ICustomerRepository customerRepository;
```

```

Administrator: Developer Command Prompt for VS2012
C:\ws\brian\ff\Dev>tf view FabrikamFiber.CallCenter\fabrikamfiber.web\controllers\customerscontroller.cs
C:\ws\brian\ff\Dev>tf view FabrikamFiber.CallCenter\fabrikamfiber.web\controllers\customerscontroller.cs /version:c7
C:\ws\brian\ff\Dev>tf view FabrikamFiber.CallCenter\fabrikamfiber.web\controllers\customerscontroller.cs /version:c26
C:\ws\brian\ff\Dev>_

```

[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

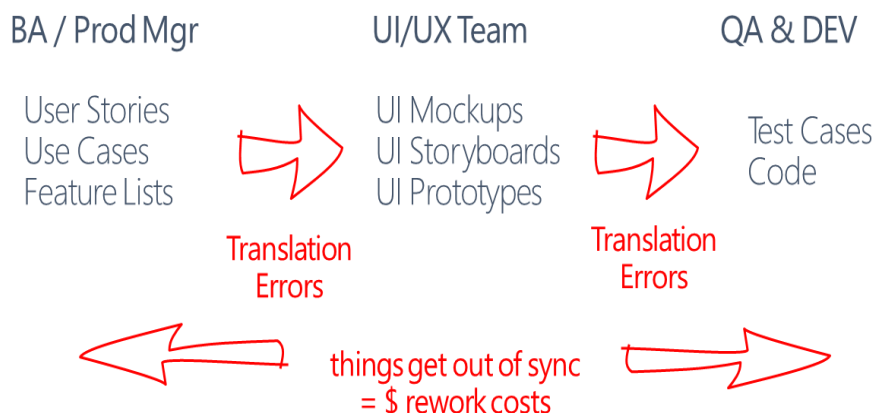
Current Approach

- 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.
3. QA and Development
 - Then the development team and the testing team need to interpret these UI Storyboards, UI Mockups and Use Cases to develop the code and Test Cases.



The result of this approach includes a lot of re-work and project costs due combinations of the following. Perhaps some of this sounds familiar to you.

- Use Cases, UI Mockups, UI Storyboards and test cases often get out of sync with each other, which makes it difficult for the developers and testers to know what real requirements are.
 - a) For example the use case steps may now communicate something different than steps outlined in the ui storyboard or test cases.
- UI Storyboards and/or prototypes forget to implement all of the flows outlined in the use cases, leading to a scenario where features get missed.
- Test cases are incorrect, causing false bugs to be found, wasting further time and budget.
- Missed test cases, causing bugs not to be uncovered, leading to costly production issues.
- Developers building features that are different than what the customer is looking for, leading to unsatisfied customers or additional rework.
- Another related issue is that often during the UI Storyboarding phase new requirements come up that are added to the UI Storyboard but also should be in the original use case. A lot of good intentioned teams will not spend the time to update the use cases and feature lists, furthering the confusion for the developers and testers.

2. What Can We Do To Reduce This Confusion And Rework

The good news is that there are some reasonably simple changes we can make in our processes and deliverables that will decrease the “broken telephone effect”. The theme shared by these ideas is to reduce the hand-offs and translations.

2.1 Get the Entire Team Involved in the Requirements Early

One technique is to involve the BA, UI/UX and QA resources directly in the elicitation process with the customer. Some may even argue to include the lead developer for that domain as well. By having all disciplines represented during the interviewing process, they will all hear requirements directly from the customer, helping reduce the reliance on just what the BA documents. Another equally important side benefit is that each discipline brings a different perspective which can lead the interview process down different paths of conversation and requirements gathering.

For example the QA resource may tend to ask more questions about the requirements that related to edge conditions, or error conditions, than perhaps the BA or UI/UX resource would. In addition putting the UI/UX resource in front of the customer, will give them a chance to better understand what kinds of features are more frequently used, which is important when trying to manage the cognitive load of the end user, by ensuring the more frequently used functionality is more easily found through good UI/UX design.

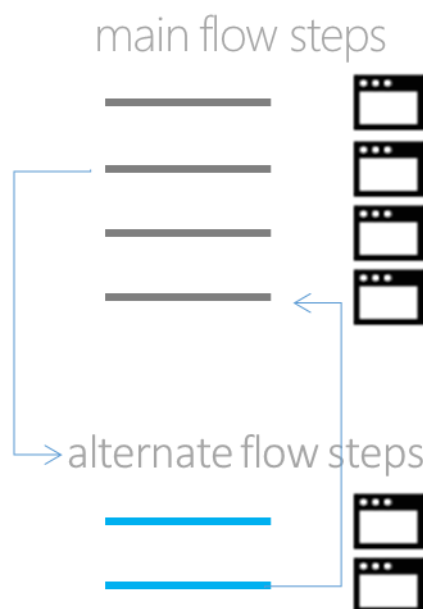
This may seem like a waste of time and budget, but at least considering the benefits of doing this on the more complex portions of your application could be a good start.

My personal approach is to include the developers and QA team during the requirements walkthroughs early and well before they are done, so that they have the opportunity to provide feedback and also gain a deeper semantic understanding of the requirements and what is driving them.

2.2 Combine and evolve your Use Cases, UI Mockups and UI Storyboards into the same physical deliverable

One approach to addressing this is to combine your Use Cases, UI mockups and UI storyboards into one integrated deliverable called a “Use Case Storyboard”. A “Use Case Storyboard” is simply a use case where each step in the use case is associated with a UI Mockup.

Use Case Storyboard



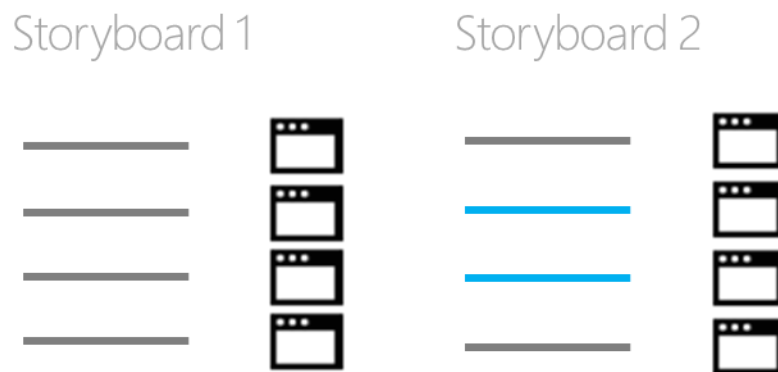
This approach has the benefit of automatically keeping what you traditionally would have documented as a separate use case, UI mockups and UI storyboards in sync. In addition to keeping everything in sync, it turns out to be far easier to communicate the requirements during walkthroughs of “use case storyboards”, because they combine the details of a use case text along with the informative UI Mockups into one deliverable that is easy to walk through from the end user point of view. This not only helps your end customer confirm the requirements are correct, but also importantly helps reduce confusion with the developers and testers.

In addition it strongly encourages the collaborative and combined review of use cases and UI/UX design by your BA and UI/UX teams, which will result in more accurately defined and understood requirements.

Adopting this approach would involve starting as we normally do by having the BA define the use case without any mention of UI specific requirements, just to capture the objective of the user in the use case and any key steps and alternate flows (or exception flows, depending on which terminology you use). Then the UI/UX team would start to attach UI Mockups to each of the steps in the Use Case to start to bring to life the UI/UX design.

Since the “Use Case Storyboard” is expressed in terms of a main flow and alternate it has the additional benefit of reducing the number of traditional storyboards, which are linear and don’t include alternate flows. For example if we took the sample “use case storyboard” above and represented it as two separate linear storyboards, the first and last steps and their UI mockups would be duplicated, leading to requirements maintenance and rework costs.

Traditional Linear Storyboards end up duplicating steps



So to summarize, the “use case storyboard” reduces rework by helping you communicate requirements more clearly. In addition this approach also reduces requirements translation errors typically caused by hand-offs between deliverables and teams.

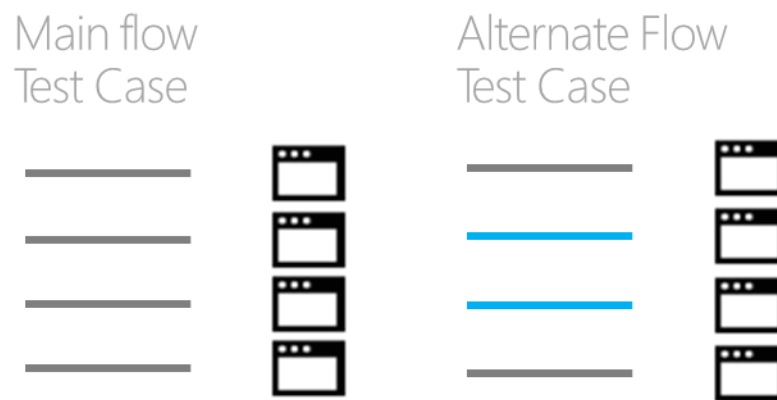
2.3 Generate Your Test Cases Automatically From Use Case Storyboards

Instead of having your QA team manually create test cases, why not Generate test cases directly from your “Use Case Storyboards”. This has the following major benefits.

1. Dramatically reduce the time it takes to create your functional test case set.
2. Your test cases are 100% consistent with your requirements
3. You have complete functional test case coverage
4. As requirements change, you can update your test case set quickly

For example if we were to take the “Use Case Storyboard” outlined above, it would generate the following two test cases, including the UI Mockups.

Test Cases



Test Cases are typically written in a format of a “user action” followed by an “expected” result. If you spend a little more time up front when creating “Use Case Storyboards”, defining which steps are “user actions” and which steps are “expected results” this can be used in the test case generation engine to determine which steps should be assembled into test cases as “user actions” or “expected results”.

3. Summary

As an industry we are always evolving how we do things. In this article I present an approach that helps reduce project costs and increase the likelihood that you will develop what your customers need. I talked about how to get your collective team involved earlier in the requirements definition process which is not really a new idea, but certainly a valuable one. However I do present the concept of a “Use Case Storyboard” that combines Use Cases, UI Mockups and UI Storyboards, into one integrated deliverable that promotes collaboration and also makes it easier to communicate requirements to your customer, developers and testers. Lastly I talked about the idea of generating your functional test cases directly from these “Use Case Storyboards” which generates some significant project savings.

[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.



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](#)

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..

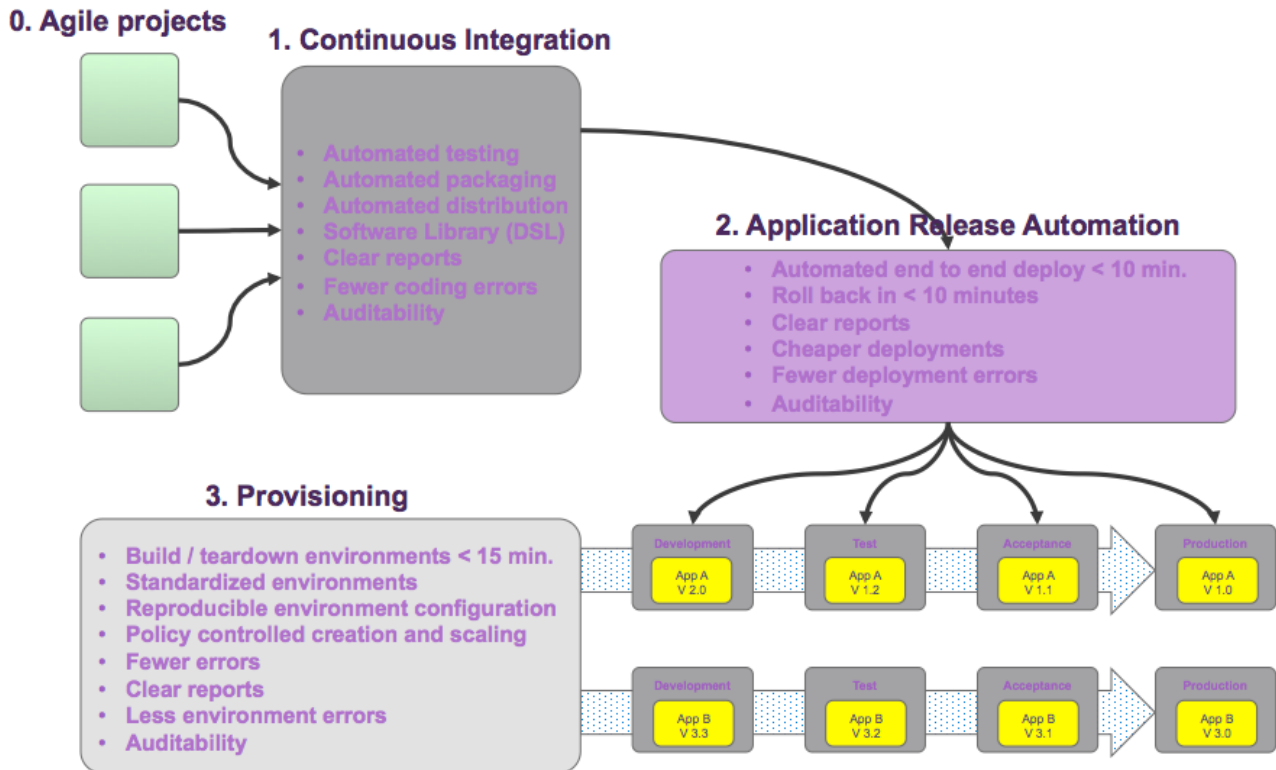


What is Continuous Delivery?

Continuous Delivery is a software delivery strategy aimed at moving your ideas into production as fast and efficiently as possible. This allows you to validate your business assumptions with the customer at a very early stage of your delivery process, allowing you to adjust when required. Your ideas will reach the customer faster than is the case with non-automated delivery environments. Because part of the solution is based upon automation of your delivery pipeline, which in itself constitutes a very error-prone process, your delivery will become more reliable.

What does automating your delivery pipeline imply?

1. Code is managed in version control.
2. Compilation and packaging of code is performed centrally in a completely automated fashion.
3. Testing is automated and is treated as an integral part of the delivery process.
4. Applications are automatically deployed, end-to-end, to one or more environments.
5. Server environments are created and torn down with a push of the button.
6. Clear, quality-focused status reports are provided based on facts, not stories.



A delivery environment that supports Continuous Delivery often consists of the following three mechanisms:

Continuous Integration – overall, this mechanism ensures on a continuous basis (i.e. whenever the code changes), that code is integrated (i.e. all bits compile and run together), tested and is made ready to deploy.

Application Release Automation – forms the bridge between the development and the operations

organization; development provides the deployment units plus configurations, where deployment happens based on automated triggers (i.e. tests passed) and is controlled by, often automated, guards or conditions. It provides the operations organization with an easy to use and standardized gateway through which development delivers its applications. It relieves operations from the need of reading through long and tedious installation manuals and configuring/loading individual components (i.e. application servers, databases, http servers, load balancers, queuing mechanisms, services busses) by hand. Deployments become a matter of minutes instead a matter of hours.

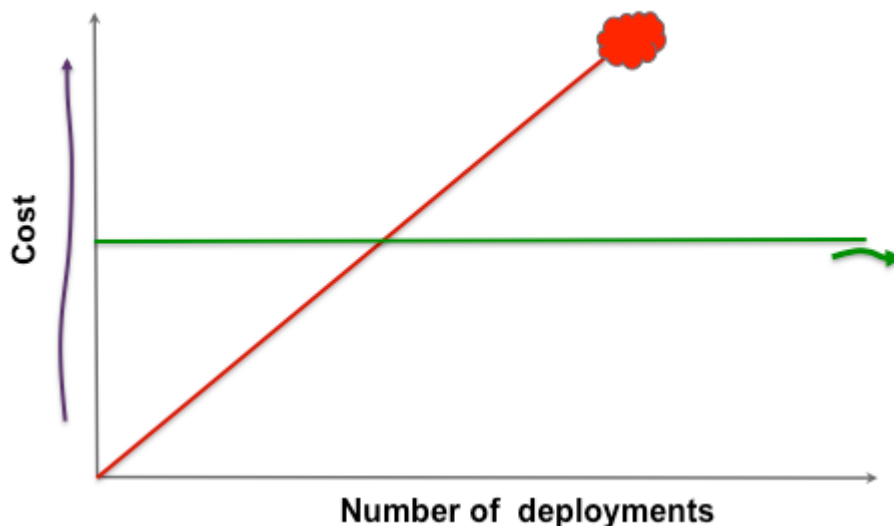
Provisioning – used by service management or the operations organization to automatically install building blocks of code to the server and properly configure the systems around it. It further ensures system reliability by enabling to deploy *known- good* applications to actual *known- good* environments. Bringing up or tearing down dedicated environments (i.e. quick test or new production environment), now becomes a matter of minutes instead of a matter of days.

What does Continuous Delivery provide your organization

It lowers your cost

Traditional: Deployments require manual tasks, expert scripting and frequent troubleshooting sessions and in that respect represent an investment. As long as the amount of deployments progress, the amount of expenditure grows. Due to rising cost (and deployment duration), the maximum amount of deployments per hour is also bound to limits.

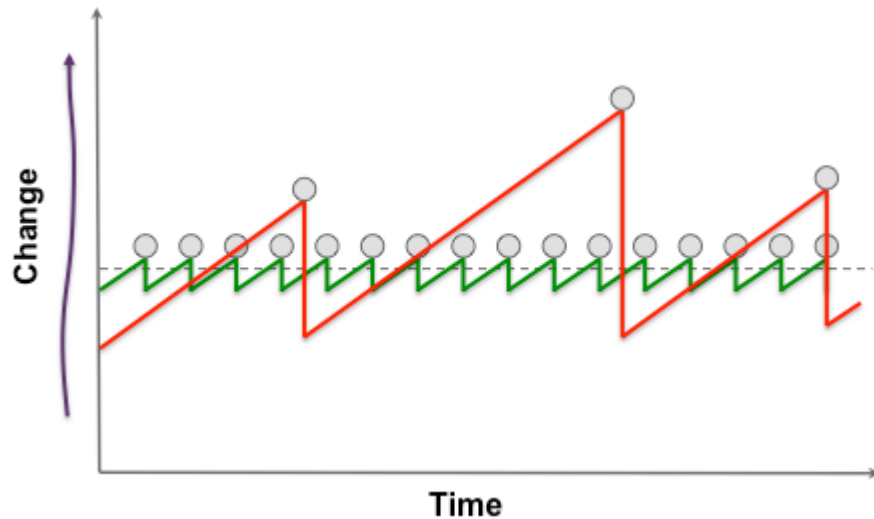
Continuous Delivery: The amount of deployments does not have a large effect on overall cost. Once a deployment pipeline is configured, subsequent deployments happen automatically or at the 'push of the button'. The maximum amount of deployments is not bound to any limit.



It shortens your time to market

Traditional: Too many changes are delivered in one big release. The time between releases is long and a large effort is required to deploy. A big release with many changes almost inevitably gets delayed because you can't get a large amount of features to work together in one go. Furthermore, the release process is not efficient: i.e. a release contains 100 changes. When 3 changes fail the test, the 97 changes that actually are correct have to wait for the 3 defects to be fixed before a move to production can be made.

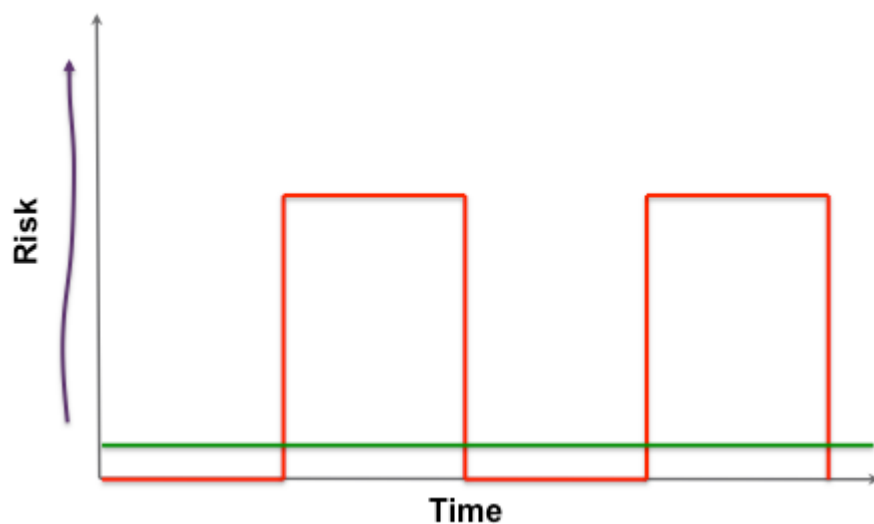
Continuous Delivery: small batches of changes are moved to production in a continuous manner and become instantly visible (if configured that way). Changes can be made immediately available to your customers and a new feature means immediate business value. Feedback from the customer can be gathered almost instantaneously. When a feature is ready for production, it can be moved to production.



It mitigates your risk

Traditional: Adding a large amount of changes into an environment introduces risk. Due to the large timeframe between releases, there is a high chance that environments also might have been changed. Every deployment becomes a “big-bang” that touches many moving parts. The chance of hitting some untested combination is high. Every deployment becomes unique, making it impossible to rely on experience from a previous deployment.

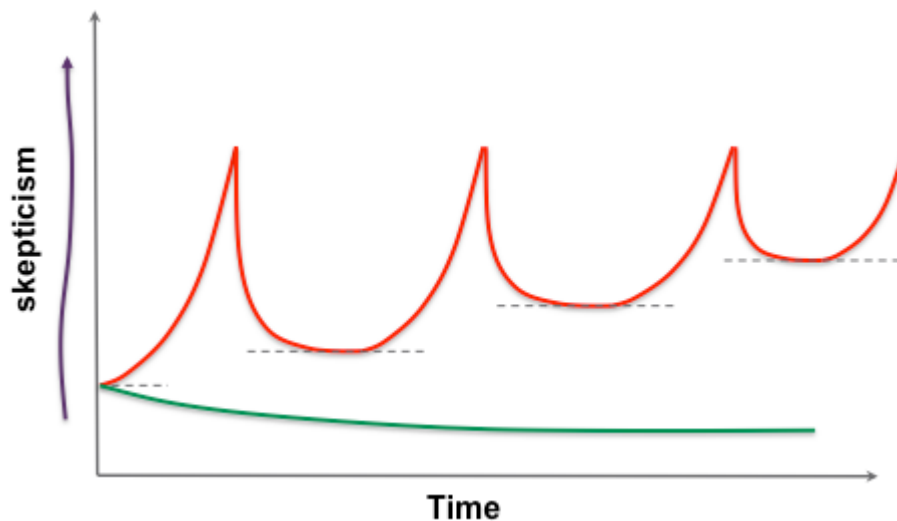
Continuous Delivery: If it hurts, do it more often. An automated deployment only has to be configured and tested once, and from thereon can be repeated many times in a row (e.g. once per hour). Deployments are proven on a continuous basis and in that respect the risk of a deployment is lower than is the case with manual, infrequent deployments. Simply put: the release process becomes far more reliable.



It (re)builds trust within your IT organization

Traditional: When an activity fails on a regular basis, skepticism grows. When a person is skeptical, this person will try to duck the responsibility or activity as a whole. The same applies to organizations. Operational organizations are responsible for continuity and by nature do not welcome change. Often a conservative approach is taken: implementing large and thorough procedures or even scheduling slack time into the planning often occurs as a measure of mitigation.

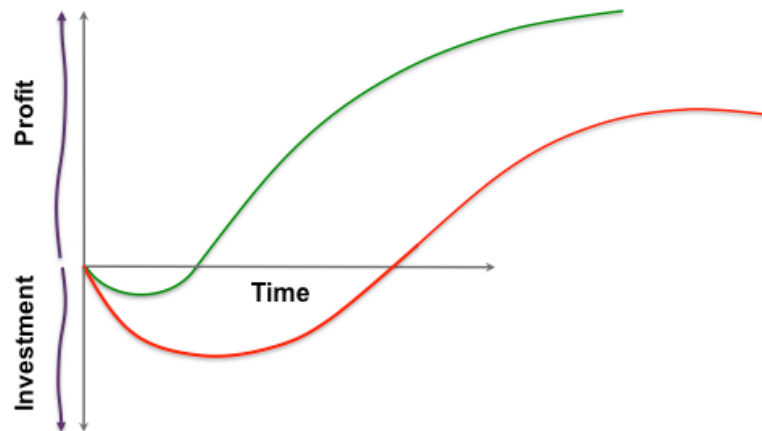
Continuous Delivery: A deployment runs completely automatically, on a daily basis and is reliable. As the deployment is proven over and over again, confidence will grow. Once the process is in place and runs correctly, there is far less reason for building up resistance between development and operations organizations. Confidence (re-)builds trust.



It helps you to understand your customer

Traditional: An idea is first fully designed and scoped before the project can progress to the task of implementing the idea itself. The result (and success) of an idea is only visible once the full investment has been made. Prior to starting a project, full budget is required and one must be fully certain that ideas are worth the investment. While the project is ongoing, the customer has no way of verifying progress or features.

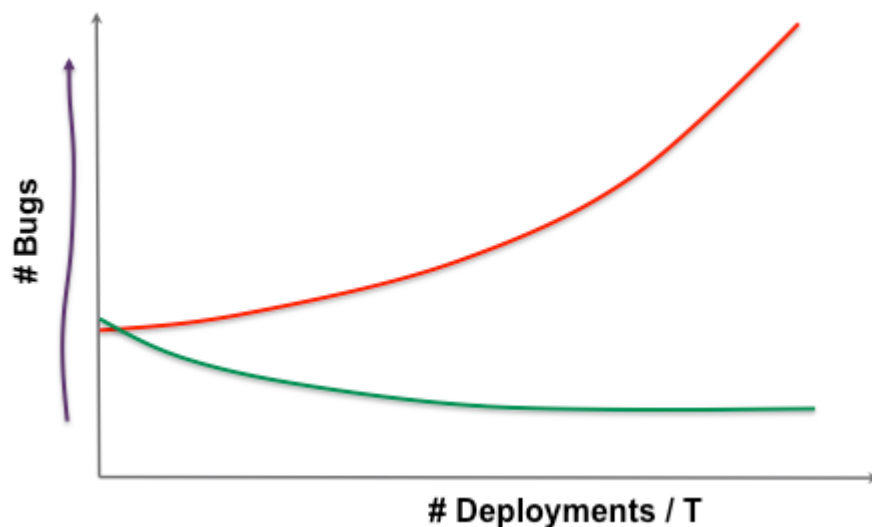
Continuous Delivery: Envision an idea, design and realize it bit-by-bit. As the project continues, investments are capitalized as features become available. The delivery process provides the possibility to adjust scope & requirements and support the decision to stop investing in features that actually appear not to be used by the end user. Because customer behavior can be measured as the product evolves, the product is better aligned with demand than is the case in traditional situations.



It raises the overall quality of your application

Traditional: Code is compiled and packaged manually whenever this is deemed necessary. Manual tests are performed once code is in its final stages, making test results visible only at the end of a project. When a test fails, it is hard to find the solution as there is no real 1:1 correlation between what was 'changed' and what needs to be fixed, which costs a lot of valuable time. The project needs to go live, so at a certain moment the code is promoted to production, even though not all issues have been solved.

Continuous Delivery: The process of assembling, compiling and testing is completely automated, forms an integral part of the delivery process and is quick and easy. Possible coding-issues become immediately visible and can be fixed accordingly when the ink is still wet. When code is checked in as ready, this code is actually really ready.



The Maturity Model

A chart measuring Continuous Delivery maturity is provided below. You can use it yourself to see where you stand and what topics require attention.

	DevOps	Reporting	Testing	Provisioning	Deploying	Building
Level 5 Complete	Operations and development are both part of the multidisciplinary delivery team and share responsibilities.	Reports also provide trend analysis.	100% fully automated tests all the way to production	Self Service portal for requesting an environment.	Continuous end-to-end deployments.	End-to-end automated gated builds.
Level 4 Advanced	An envoy of operations works along in project, an envoy of development works along with operations.	Dashboard provides insight from different perspectives and shows history and progression through a build monitor to all.	Automated dynamic quality tests like security scans, functional and performance tests guarantee quality of code.	Environments can be created and torn down by a push of the button. Operating System is virtualized.	Test-gated deployments of end-to-end applications. Deployments occur over multiple environments.	Central build environment. Teams actively reuse generic components in a secure and controlled manner.
Level 3 Average	Development and operations work together when this is required.	Graphical and textual reports accessible through dashboard.	Automated static code and security analysis after code check in.	Environments are identical. Several tools used to provision and configure an environment.	Environments are identical. Roll out of applications performed by a push of the button. Auto-deployment to D, T, A and P.	Build on commit. Archived components are made available for reuse by other teams.
Level 2 Beginner	Code accompanied with release notes with which operations should install and manage the application.	Reports generated on request by system administrator. Reports are graphical in nature.	Automated tests are initiated as soon as code is checked in. Tests are focused on unit/component testing only.	Scripted installations per component for each server. Surrounding systems manually configured.	Self service deployments to development and test.	Automated builds are performed in a central area and activated manually.
Level 1 Base	Operations engaged at the end of the project.	Reports generated on request by system administrator. Reports are text based.	All tests require manual activity. Some tests are automated but have to be initiated by hand.	Manual installation and configuration of software for middleware, databases, application servers, etc.	Deployment through execution of separate deployment -and db scripts. Manual configurations and installs / env.	Builds are performed on local workstation by use of one or more separate build scripts.

DevOps

In what manner do operations and development work with one another and what is the structure of the complete delivery team on an overall level? Are applications moved from one organization to another, or do disciplines work well together, define a real team and feel of end-to-end responsibility?

Reporting

In what way or manner can we gain actual insight into the state of the projects? Are status reports created manually and do they only report on the items that people are actually aware of, or is there a dashboard providing actual status (i.e. provides a holistic view) on *all* aspects one needs to be aware of?

Testing

What is the attitude towards testing? Are tests performed when a project almost finishes or are tests automatically performed at the start of the delivery process, guaranteeing quality as early as possible? Is testing treated as an integral aspect of a development project and is most of it automated?

Provisioning

In what manner are the systems in which your application runs configured? Is this done manually, introducing the risk of configuration drift, misconfigurations and time-consuming mistakes, or is this configured as a repeatable process with which you install version-controlled, standardized platforms in a matter of minutes?

Deploying

When deploying and configuring end-to-end applications across multiple servers, are these activities performed manually, or is this implemented as an automated, repeatable and predictable process based on Application Release Automation (ARA)?

Building

Are builds performed locally on the developer's workstation or at regular intervals, in a central place, tested, while making the resulting components accessible for reuse by other teams?

The Principles

When considering a move to Continuous Delivery, some basic software development process principles apply:

$$\text{higher quality} + \text{automation}^{(\text{test} + \text{provisioning} + \text{deployment})} = \text{cost reduction} + \text{acceleration}$$

Higher Quality

Applying quality mechanisms from the very start of your software delivery process will alleviate the potential for costly issues to be revealed only at the end of the project, when the pressure is high and software needs to go live. Raising the bar on quality requires attention to automated testing features, standardization of your environment and minimizing the amount of human interactions that are required for creating and tearing down (test) environments.

Automation test

Over the last couple of years, major improvements have been made in test automation. Every type of test – functional, performance, stress - can be automated, allowing test activities to be treated as an integral part of your software delivery process. Automated tests will become faster and more consistent than would be the case if these tests were executed manually and can be executed at lower cost.

Automation provisioning

Constructing and tearing down the Development, Testing, Acceptance and Production (D, T, A, P) environments is a highly error-prone part of the software delivery process. A totally manual install and configuration of software, middleware, databases and servers requires such a large number of steps that it is nearly impossible to get right in one go, and especially to get right consistently across environments.

Automating the provisioning of your system will allow you to construct and tear down environments with the push of a button, making this activity faster, less complex, less costly and far less error prone.

Automation deployment

Another error-prone aspect of the software delivery process is the actual deployment of applications. Using a tool for Application Release Automation (ARA) will enable you to deploy and redeploy applications over your D, T, A and P environment in a matter of minutes, over and over again. Long and tedious installation manuals are replaced by standard instruction sets (templates), moving you to a state of errorless deployments that can be performed on a repetitive basis. This, of course, will save you a lot of unnecessary time and costs!

Cost reduction

As long as your software installation and application deployments process are not automated, the delivery and installation of each new D, T, A and P environment will cost you a large portion of your budget. By standardizing these processes and applying them automatically across all your environments, you invest only once and benefit from this investment over and over again. Add to that the possibility of tearing down environments at any time when not in use, and you will improve the operational effectiveness of your organization as you go.

Apart from making your operational environment more reliable, automated testing will also lower your costs. Expensive errors or bugs will not pop up close to the end of your project, or even after go-live, but at the beginning, when repair is comparatively cheap and causes still easy to find. This allows your teams to no longer have to focus on fixing costly issues but instead on what feels productive: creating software.

Acceleration

Through automation and a far less error-prone software delivery process, you will be able to deliver new functionality to production on a continuous basis. The amount of time-consuming manual activities are reduced to a minimum and, by taking an integrated and automated test approach, fewer errors will occur all the way at the end, resulting in faster feature delivery.

All in all, provisioning your environments, testing your software and then deploying your applications will become an activity of hours or minutes, not days or even weeks.!

Steps to Getting Started

Assessment

Take a quick assessment to gain a clear picture of the current status of your delivery processes. Assess the status of topics identified by the Continuous Delivery maturity model. More detail on this model is provided in the next chapter. The assessment is essential to determine order, approach and expected timelines for implementing Continuous Delivery.

Engage the development team

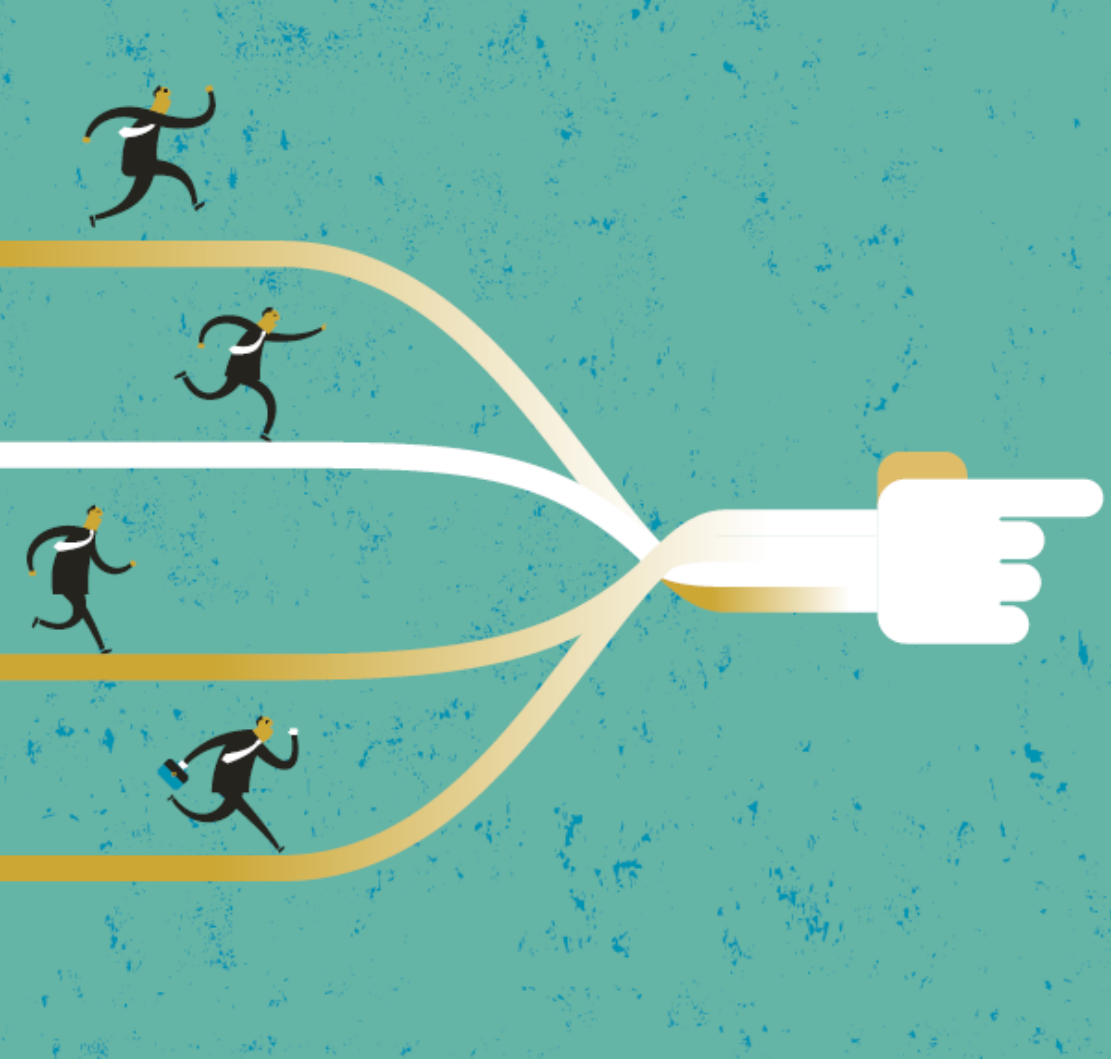
When a plan has been determined, engage with your development team. Discuss the approach and engage them in implementing a new, fresh and automated delivery process. Experience shows that development teams love to upgrade their way of working, so this is the nice part. Encourage the development team to participate in determining what is needed in this new environment and especially where they think the current process can be optimized. Define a strategy and way of working and utilize Value Stream Analysis (VSA) to see where the greatest benefits can be found.

Implement and align tools & processes

This is the phase in which configuration management systems are consolidated, build and automated test tools are aligned, the Definitive Software Library (DSL) is introduced and Application Release Automation (ARA) mechanisms are added. Concepts and guidelines are explained and written out, ensuring proper use for subsequent development teams that will start utilizing this new approach. In this phase, processes and guidelines are optimized.

Formalize the environment

Once the new tools and principles are in place, it is time to introduce other development-related initiatives to the new Continuous Delivery environment too. Make sure short and descriptive explanations of the new process are available and guide people in applying them



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

Formalize the environment

Once the new tools and principles are in place, it is time to introduce other development-related initiatives to the new Continuous Delivery environment too. Make sure short and descriptive explanations of the new process are available and guide people in applying them correctly. This phase might take some time, as people have to grow into the new way of working. Slowly step away from release-like thinking for the projects that utilize this new environment. Move away from big-bang releases towards gradually bringing new functionality into production.

Achieve Continuous Delivery

The final phase is defined to remove all roadblocks standing in the way of actually applying Continuous Delivery as the standard in your organization. Actual Continuous Delivery is achieved when all involved are capable of delivering new functionality in an agile manner. This final phase is aimed at further streamlining all parties involved: from business all the way to operations, everybody has to share a product-like mindset and feel responsible for the end product. For example: Break the build = fix the build, meaning if one leaves the code broken, one fixes the code before going home as code that is in the pipeline has to be potentially shippable at all times.

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](#) (Microsoft login required) 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

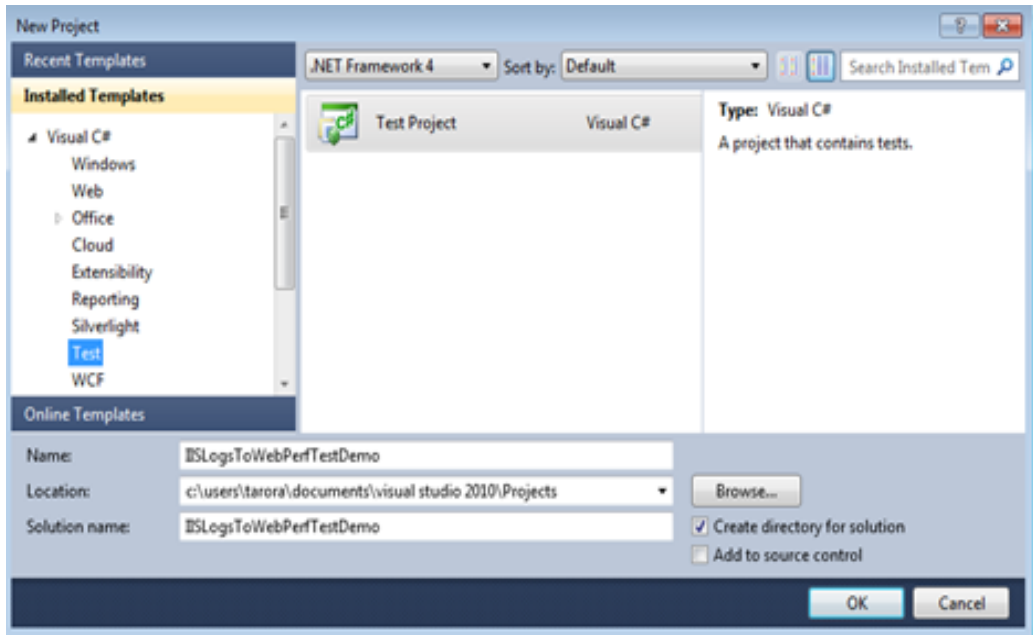
To help with server use and analysis, IIS is integrated with several types of log files. These log file formats provide information on a range of websites and specific statistics, including Internet Protocol (IP) addresses, user information and site visits as well as dates, times and queries. If you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

Walkthrough

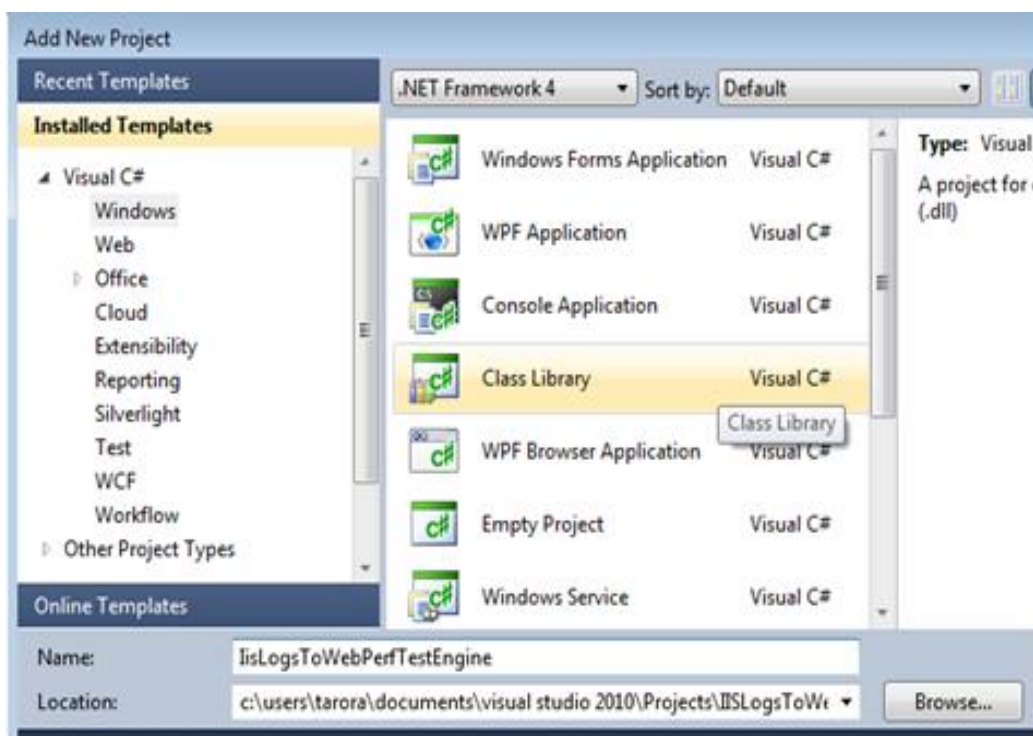
1. Download and Install Log Parser from the Microsoft [download](#) Center. You should see the LogParser.dll in the install folder, the default install location is `C:\Program Files (x86)\Log Parser 2.2`. LogParser.dll gives us a library to query the iis log files programmatically. By the way if you haven't used Log Parser in the past, it is a powerful, versatile tool

that provides universal query access to text-based data such as log files, XML files and CSV files, as well as key data sources on the Windows operating system such as the Event Log, the Registry, the file system, and Active Directory. [More details...](#)

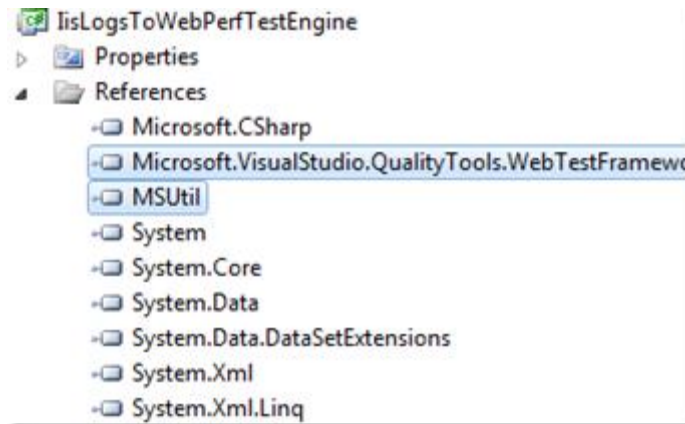
2. Create a new test project in Visual Studio. Let's call it *IISLogsToWebPerfTestDemo*.



3. Delete the `UnitTest1.cs` class that gets created by default. Right click the solution and add a project of type `Class Library`, name it, *IISLogsToWebPerfTestEngine*. Delete the default class `Program.cs` that gets created with the project.



4. Under the *IISLogsToWebPerfTestEngine* project add a reference to



- Microsoft.VisualStudio.QualityTools.WebTestFramework – c:\Program Files (x86)\Microsoft Visual Studio10.0\Common7\IDE\PublicAssemblies\Microsoft.VisualStudio.QualityTools.WebTestFramework.dll
- LogParser also called MSUtil - c:\users\tarora\documents\visual studio 2010\Projects\IISLogsToWebPerfTest\IISLogsToWebPerfTestEngine\obj\Debug\Interop.MSUtil.dll

5. Right click IISLogsToWebPerfTestEngine project and add a new class – IISLogReader.cs
The IISLogReader class queries the iis logs using the log parser.

```
using System;
using System.Collections.Generic;
using System.Text;
using MSUtil;
using LogQuery = MSUtil.LogQueryClassClass;
using IISLogInputFormat = MSUtil.COMIISW3CInputContextClassClass;
using LogRecordSet = MSUtil.ILogRecordset;
using Microsoft.VisualStudio.TestTools.WebTesting;
using System.Diagnostics;
```

```
namespace IISLogsToWebPerfTestEngine
{
    // By making use of log parser it is possible to query the iis log using select queries
    public class IISLogReader
    {
        private string _iisLogPath;

        public IISLogReader(string iisLogPath)
        {
            _iisLogPath = iisLogPath;
        }
    }
}
```

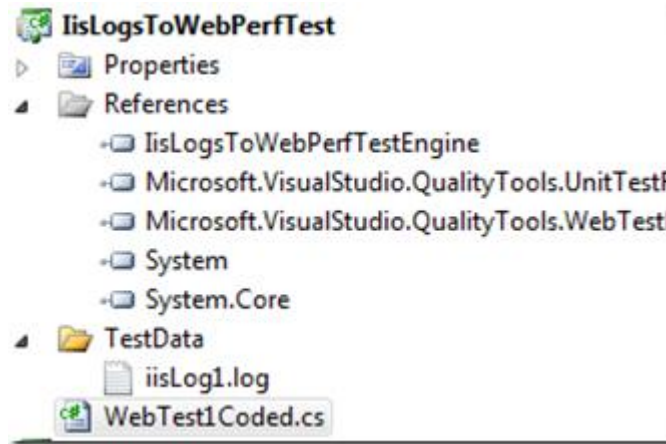
```
public IEnumerable<WebTestRequest> GetRequests()
{
    LogQuery logQuery = new LogQuery();
    IISLogInputFormat iisInputFormat = new IISLogInputFormat();

    // currently these columns give us sufficient information to construct the web test requests
    string query = @"SELECT s-ip, s-port, cs-method, cs-uri-stem, cs-uri-query FROM " +
        _iisLogPath;
    LogRecordSet recordSet = logQuery.Execute(query, iisInputFormat);

    // Apply a bit of transformation
    while (!recordSet.atEnd())
    {
        ILogRecord record = recordSet.getRecord();
        if (record.getValueEx("cs-method").ToString() == "GET")
        {
            string server = record.getValueEx("s-ip").ToString();
            string path = record.getValueEx("cs-uri-stem").ToString();
            string querystring = record.getValueEx("cs-uri-query").ToString();

            StringBuilder urlBuilder = new StringBuilder();
            urlBuilder.Append("http://");
            urlBuilder.Append(server);
            urlBuilder.Append(path);
            if (!String.IsNullOrEmpty(querystring))
            {
                urlBuilder.Append("?");
                urlBuilder.Append(querystring);
            }
            // You could make substitutions by introducing parameterized web tests.
            WebTestRequest request = new WebTestRequest(urlBuilder.ToString());
            Debug.WriteLine(request.UrlWithQueryString);
            yield return request;
        }
        recordSet.moveToNext();
    }
    Console.WriteLine(" That's it! Closing the reader");
    recordSet.close();
}
}
```

6. Connect the dots by adding the project reference 'IisLogsToWebPerfTestEngine' to 'IisLogsToWebPerfTest'. Right click the 'IisLogsToWebPerfTest' project and add a new class 'WebTest1Coded.cs'



The WebTest1Coded.cs inherits from the WebTest class. By overriding the GetRequestMethod we can inject the log files to the IISLogReader class which uses Log parser to query the log file and extract the web requests to generate the web test request which is yielded back for play back when the test is run.

```
namespace IisLogsToWebPerfTest
```

```
{
    using System;
    using System.Collections.Generic;
    using System.Text;
    using Microsoft.VisualStudio.TestTools.WebTesting;
    using Microsoft.VisualStudio.TestTools.WebTesting.Rules;
    using IisLogsToWebPerfTestEngine;
```

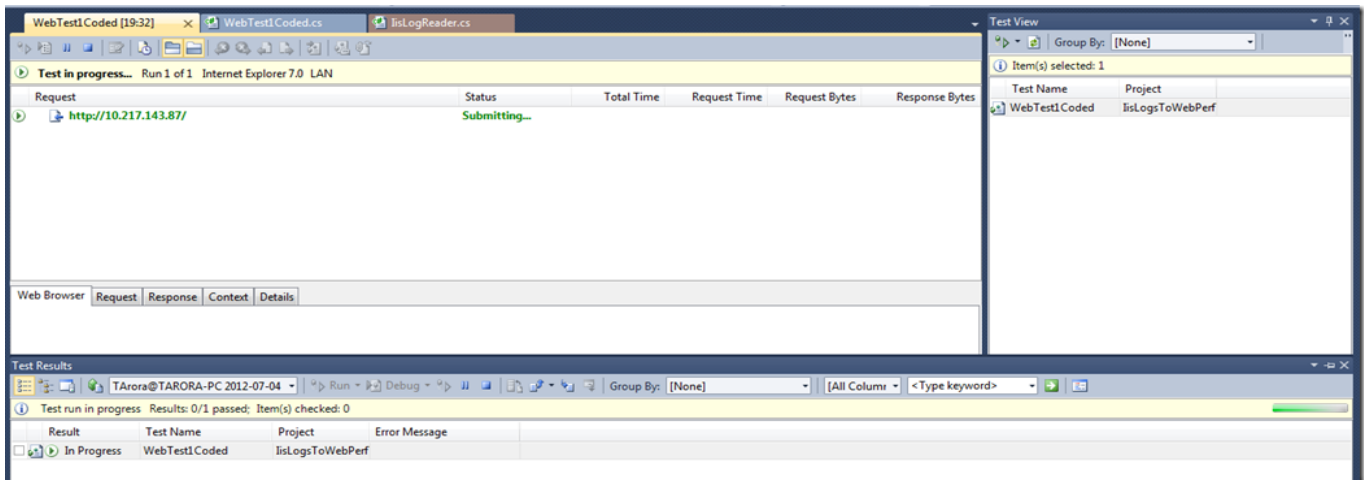
```
// This class is a coded web performance test implementation, that simply passes
// the path of the iis logs to the IisLogReader class which does the heavy
// lifting of reading the contents of the log file and converting them to tests.
```

```
// You could have multiple such classes that inherit from WebTest and implement
// GetRequestEnumerator Method and pass differnt log files for different tests.
```

```
public class WebTest1Coded : WebTest
{
    public WebTest1Coded()
    {
        this.PreAuthenticate = true;
    }
}
```

```
public override IEnumerator<WebTestRequest> GetRequestEnumerator()
{
    // substitute the highlighted path with the path of the iis log file
    IISLogReader reader = new IISLogReader(@"C:\Demo\iisLog1.log");
    foreach (WebTestRequest request in reader.GetRequests())
    {
        yield return request;
    }
}
}
```

7. Its time to fire the test off and see the iis log playback as a web performance test. From the Test menu choose Test View Window you should be able to see the WebTest1Coded test show up. Highlight the test and press Run selection (you can also debug the test in case you face any failures during test execution).



8. Optionally you can create a Load Test by keeping 'WebTest1Coded' as the base test.

Conclusion

You have just helped your testing team, you now have become the coolest developer in your organization! Jokes apart, log parser and web performance test together allow you to save a lot of time by not having to worry about what to test or even worrying about how to record the test. If you haven't already, download the solution from [here](#) (Microsoft login required). You can take this to the next level by using LogParser to extract the log files as part of an end of day batch to a database. See the usage trends by user this solution over a longer term and have your tests consume the web requests now stored in the database to generate the web performance tests.

Keep RockiNg!

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		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

TY	LY	Title / Author		Year
9	6	Clean Code: A Handbook of Agile Software Craftsmanship	Robert C. Martin	2008
10	8	Refactoring: Improving the Design of Existing Code	Martin Fowler, et al.	1999
11	3	Working Effectively with Legacy Code	Michael Feathers	2004
12	10	The Pragmatic Programmer: From Journeyman to Master	Andrew Hunt, David Thomas	1999
13	50	Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation	Jez Humble, David Farley	2010
14	12	Growing Object-Oriented Software, Guided by Tests	Steve Freeman, Nat Pryce	2009
15	4	Kanban: Successful Evolutionary Change for Your Technology Business	David J. Anderson	2010
16	15	Scrum and XP from the Trenches	Henrik Kniberg	2007
17	-	Agile in a Flash: Speed-Learning Agile Software Development *	Jeff Langr, Tim Ottinger	2011
18	11	User Stories Applied: For Agile Software Development	Mike Cohn	2004
19	34	The Clean Coder	Robert C. Martin	2011
20	17	Domain-Driven Design: Tackling Complexity in the Heart of Software	Eric Evans	2003
21	14	The Art of Agile Development	James Shore, Shane Warden	2007
22	13	The Principles of Product Development Flow: Second Generation Lean Product Development	Donald G. Reinertsen	2009
23	45	Management 3.0: Leading Agile Developers, Developing Agile Leaders	Jurgen Appelo	2011
24	16	Lean Software Development: An Agile Toolkit	Mary Poppendieck, Tom Poppendieck	2003
25	22	Making Things Happen: Mastering Project Management	Scott Berkun	2008
26	44	The Elements of Scrum	Chris Sims, Hillary Louise Johnson	2011

TY	LY	Title / Author		Year
27	29	xUnit Test Patterns: Refactoring Test Code	Gerard Meszaros	2007
28	93	Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise	Dean Leffingwell	2010
29	18	Agile Principles, Patterns, and Practices in C#	Robert C. Martin, Micah Martin	2006
30	33	Managing the Design Factory	Donald G. Reinertsen	1997
31	20	Implementing Lean Software Development: From Concept to Cash	Mary Poppendieck, Tom Poppendieck	2006
32	26	Agile Coaching	Rachel Davies, Liz Sedley	2009
33	21	Practices of an Agile Developer: Working in the Real World	Venkat Subramaniam, Andy Hunt	2005
34	-	Test Driven Development for Embedded C *	James Grenning	2011
35	69	Bridging the Communication Gap: Specification by Example and Agile Acceptance Testing	Gojko Adzic	2009
36	24	Behind Closed Doors: Secrets of Great Management	Johanna Rothman, Esther Derby	2005
37	35	Agile Retrospectives: Making Good Teams Great	Esther Derby, Diana Larsen	2006
38	25	Crystal Clear: A Human-Powered Methodology for Small Teams	Alistair Cockburn	2004
39	31	Writing Effective Use Cases	Alistair Cockburn	2000
40	27	Applied Software Project Management	Andrew Stellman, Jennifer Greene	2005
41	-	The Culture Game: Tools for the Agile Manager	Dan Mezick	2012
42	48	The Concise Executive Guide to Agile	Israel Gat	2010
43	30	Manage Your Project Portfolio: Increase Your Capacity and Finish More Projects	Johanna Rothman	2009
44	19	Agile Testing: A Practical Guide for Testers and Agile Teams	Lisa Crispin, Janet Gregory	2009

TY	LY	Title / Author		Year
45	38	Refactoring to Patterns	Joshua Kerievsky	2004
46	40	The Productive Programmer	Neal Ford	2008
47	39	Extreme Programming Explained: Embrace Change (1st+2nd Edition)	Kent Beck, Cynthia Andres	1999
48	42	Agile and Iterative Development: A Manager's Guide	Craig Larman	2003
49	62	Innovation Games: Creating Breakthrough Products Through Collaborative Play	Luke Hohmann	2006
50	92	Agile Game Development with Scrum	Clinton Keith	2010
51	58	Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition	Lyssa Adkins	2010
52	-	Lessons in Agile Management: On the Road to Kanban	David J. Anderson	2012
53	36	Agile Project Management with Scrum	Ken Schwaber	2004
54	51	Requirements by Collaboration	Ellen Gottesdiener	2002
55	97	Agile Software Development with Distributed Teams	Jutta Eckstein	2010
56	-	Agile Project Management For Dummies	Mark C. Layton	2012
57	46	Test Driven Development: By Example	Kent Beck	2002
58	52	Manage It!: Your Guide to Modern, Pragmatic Project Management	Johanna Rothman	2007
59	28	Agile Project Management: Creating Innovative Products (2nd Edition)	Jim Highsmith	2009
60	55	Leading Lean Software Development: Results Are not the Point	Mary Poppendieck, Tom Poppendieck	2009
61	-	Lean from the Trenches: Managing Large-Scale Projects with Kanban	Henrik Kniberg	2011
62	47	Agile Software Development with Scrum	Ken Schwaber, Mike Beedle	2001

TY	LY	Title / Author		Year
63	54	Organizational Patterns of Agile Software Development	James O. Coplien, Neil B. Harrison	2004
64	-	The Power of Scrum	Jeff Sutherland, Rini van Solingen, Eelco Rustenburg	2011
65	-	Liftoff: Launching Agile Teams & Projects	Diana Larsen, Ainsley Nies	2011
66	37	Agile Adoption Patterns: A Roadmap to Organizational Success	Amr Elssamadisy	2008
67	41	Agile Product Management with Scrum: Creating Products that Customers Love	Roman Pichler	2010
68	49	Continuous Integration: Improving Software Quality and Reducing Risk	Paul M. Duvall, Steve Matyas, Andrew Glover	2007
69	56	Ship it! A Practical Guide to Successful Software Projects	Jared Richardson, William A. Gwaltney	2005
70	99	Scaling Software Agility: Best Practices for Large Enterprises	Dean Leffingwell	2007
71	59	Collaboration Explained: Facilitation Skills for Software Project Leaders	Jean Tabaka	2006
72	60	Beyond Software Architecture: Creating and Sustaining Winning Solutions	Luke Hohmann	2003
73	78	Agile Excellence for Product Managers: A Guide to Creating Winning Products with Agile Development Teams	Greg Cohen	2010
74	43	Stand Back and Deliver: Accelerating Business Agility	Pollyanna Pixton, Niel Nickolaisen, Todd Little, Kent McDonald	2009
75	61	Changing Software Development: Learning to Become Agile	Allan Kelly	2008
76	76	The Software Project Manager's Bridge to Agility	Michele Sliger, Stacia Broderick	2008
77	57	Kanban and Scrum - Making the Most of Both	Henrik Kniberg, Mattias Skarin	2010

TY	LY	Title / Author		Year
78	-	Essential Scrum: A Practical Guide to the Most Popular Agile Process	Kenny Rubin	2012
79	94	The Enterprise Unified Process: Extending the Rational Unified Process	Scott W. Ambler, John Nalbone, Michael J. Vizdos	2005
80	23	Beautiful Testing: Leading Professionals Reveal How They Improve Software	Adam Goucher, Tim Riley	2009
81	53	Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum	Craig Larman, Bas Vodde	2008
82	63	Just Enough Requirements Management: Where Software Development Meets Marketing	Alan Mark Davis	2005
83	91	Personal Kanban: Mapping Work Navigating Life	Jim Benson, Tonianne DeMaria Barry	2011
84	66	Extreme Programming Installed	Ron Jeffries, Ann Anderson, Chet Hendrickson	2000
85	67	Beautiful Teams: Inspiring and Cautionary Tales from Veteran Team Leaders	Andrew Stellman, Jennifer Greene	2009
86	65	Implementation Patterns	Kent Beck	2006
87	64	Agility and Discipline Made Easy: Practices from OpenUP and RUP	Per Kroll, Bruce MacIsaac	2006
88	68	Refactoring Databases: Evolutionary Database Design	Scott W. Ambler, Pramodkumar J. Sadalage	2006
89	75	Test Driven: TDD and Acceptance TDD for Java Developers	Lasse Koskela	2007
90	71	Agile Software Development: The Cooperative Game (2nd Edition)	Alistair Cockburn	2006
91	74	Emergent Design: The Evolutionary Nature of Professional Software Development	Scott L. Bain	2008
92	73	Becoming Agile: ...in an Imperfect World	Greg Smith, Ahmed Sidky	2008
93	79	Managing Agile Projects	Kevin J. Aguanno	2005

TY	LY	Title / Author		Year
94	-	Software in 30 Days: How Agile Managers Beat the Odds...	Ken Schwaber, Jeff Sutherland	2012
95	-	Essential Skills for the Agile Developer	Alan Shalloway, Scott Bain, Ken Pugh, Amir Kolsky	2011
96	70	Managing Agile Projects	Sanjiv Augustine	2005
97	-	Business Patterns for Software Developers	Allan Kelly	2012
98	-	Adapting Configuration Management for Agile Teams: Balancing Sustainability and Speed	Mario E. Moreira	2009
99	-	Agile Software Engineering with Visual Studio: From Concept to Continuous Feedback (2nd Edition)	Sam Guckenheimer, Neno Loje	2011
100	-	Agile ALM: Lightweight Tools and Agile Strategies	Michael Hüttermann	2011

TY = position this year

LY = position last year

*) UPDATE 24-8-2012: Added to the list one week after original publication

This list is based on **quality** (average ratings) and **quantity** (number of ratings), both on [Amazon.com](#) and [GoodReads.com](#). The age of the books also played a minor factor in the calculations. (Older books should keep acquiring new ratings, or else they drop in the list.)

Basically, I created four different rankings: (1) average Amazon rating, (2) number of Amazon ratings, (3) average GoodReads rating, (4) number of GoodReads ratings, (5) age of book in days since release. Then I calculated the average score of the books across these rankings (where age had only half the weight compared to the others), and then I turned this into the final ranking. The result is a list that reflects books with the most and the best ratings, weighed according to the time span in which they were able to earn those ratings.

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.

ALM Mag.com

Empowering Rapid Delivery of Quality Software

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