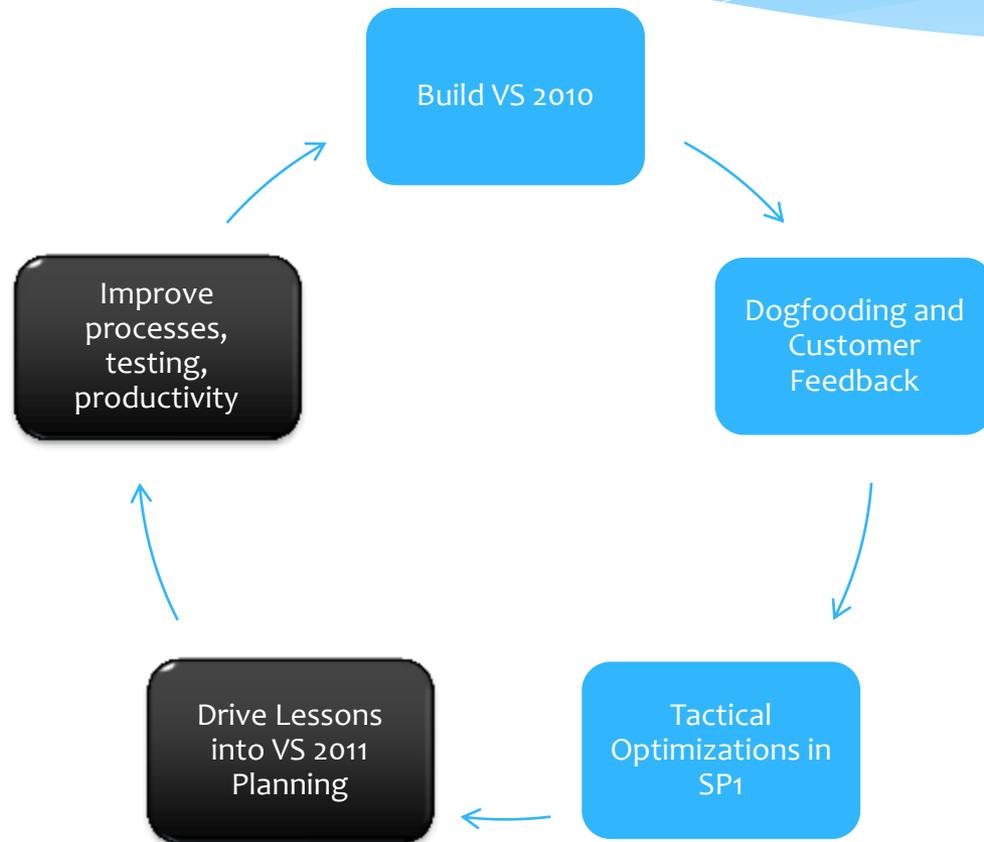


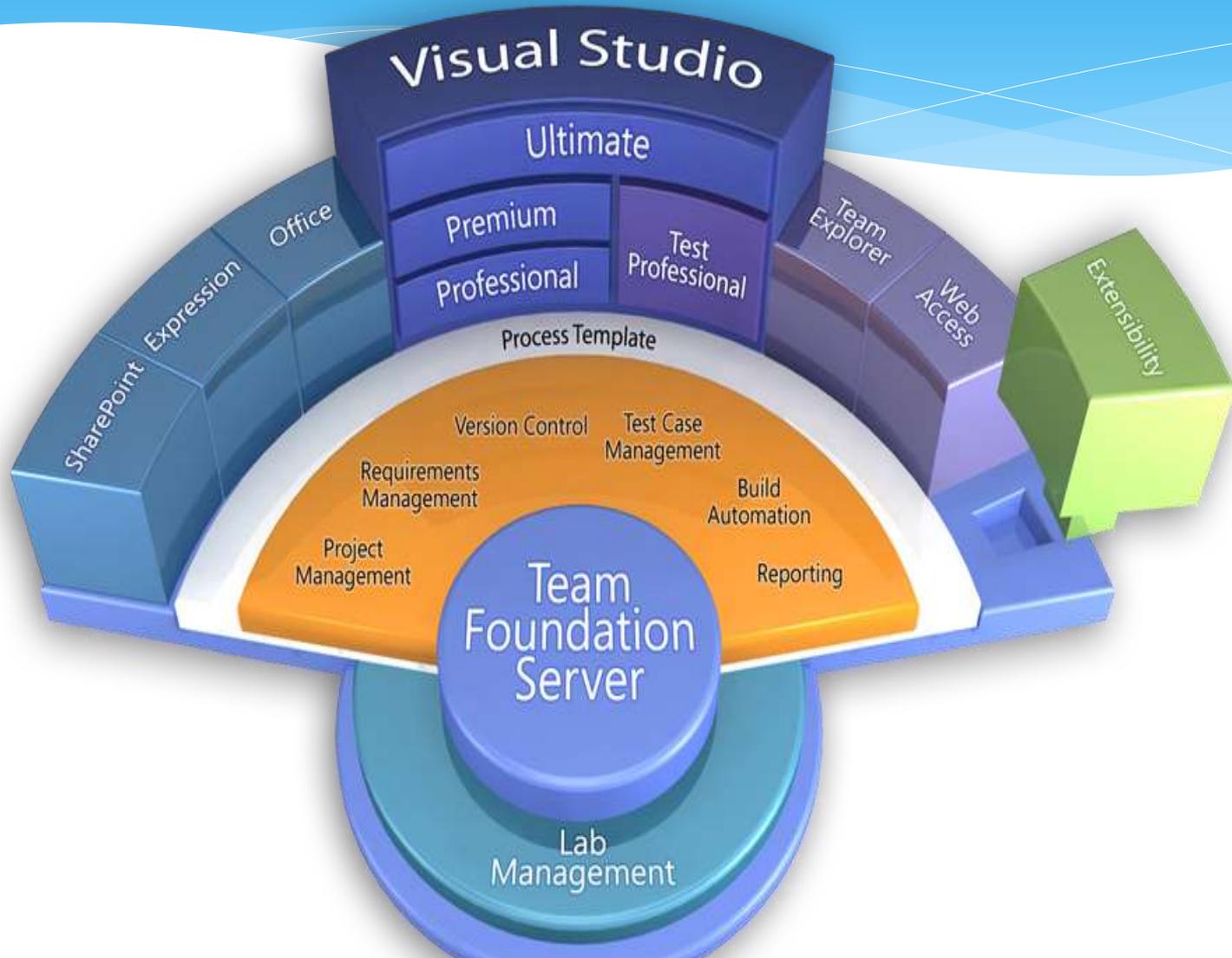
# From Lessons Learned to Lessons Productized

Dr. Tim Wagner  
Microsoft Visual Studio  
VS Ultimate Director of Development  
QCon 2010, SF

# Feedback Loop



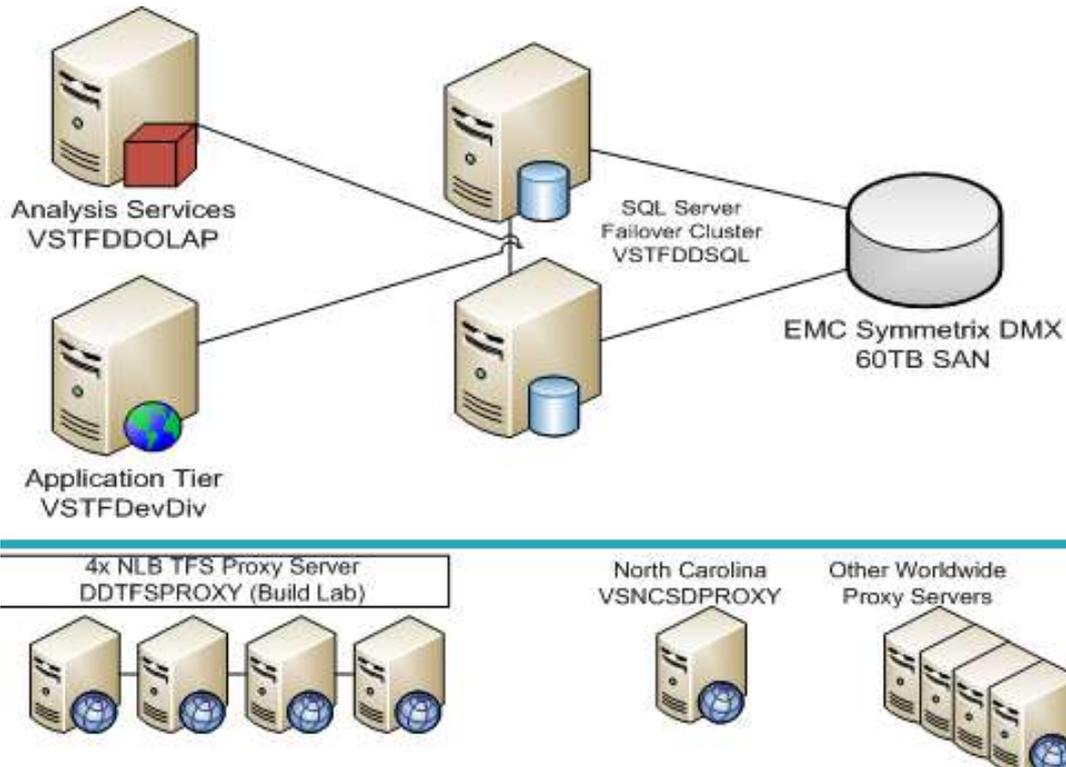
# A 2008 Example: Team Foundation Server Performance



# Dogfood? Really?

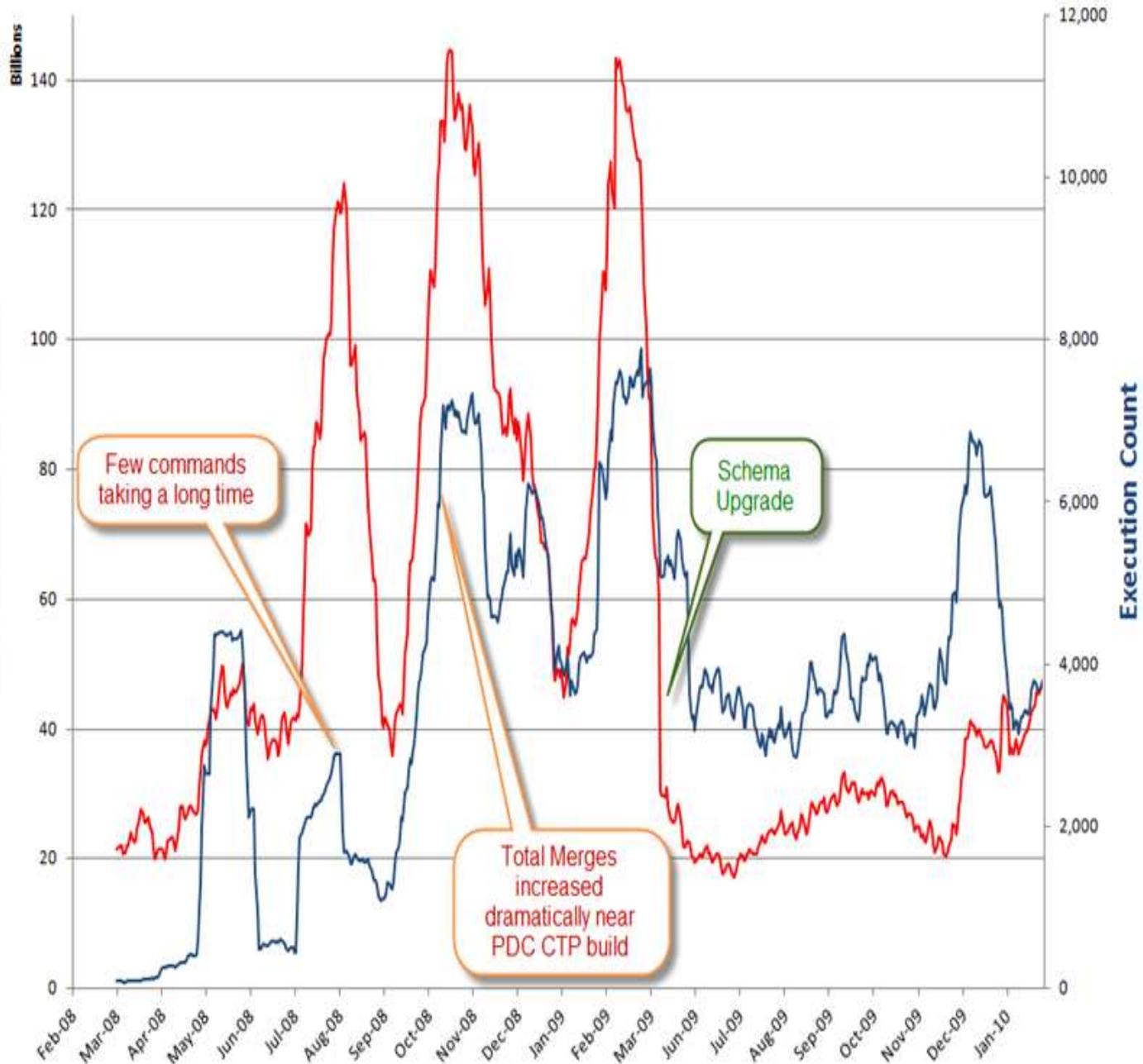


# How much dogfood?



- \* Database: 10 TB
- \* Users: 3,481
- \* Files: 1,033,167,658
- \* Uncompressed File Sizes: ~16TB
- \* Checkins: 2,047,024
- \* Shelvesets: 265,150
- \* Merge History: 2,458,112,813
- \* Pending Changes: 29,745,648
- \* Workspaces: 41,466
- \* Total Work Items: 913,619
- \* **Last 30 days...**
  - \* Work Item queries: 275,806
  - \* Work Item updates: 21,112
  - \* Checkins: 20,975
  - \* Shelves: 10,899
  - \* Gets: 410,540

Execution Time (seconds)



Few commands taking a long time

Schema Upgrade

Total Merges increased dramatically near PDC CTP build

# Lessons Learned

- \* The worse the pain, the more you need to feel it.
- \* You can't simulate problems of scale.
  - \* 99% uptime for 400 is fine... 99% uptime for 4,000 is not
  - \* Problems of heterogeneity only manifest with a sufficiently large population

# Stories from Visual Studio 2010...

- \* Gee, that looks scary– scaling successfully
- \* Untangling spaghetti – architectural dependencies
- \* Where are my reading glasses – a cautionary UI tale
- \* Dirty laundry – software components behaving badly

Caveat: This is not a product preview.

# VS 2010: Gee, That Looks Big

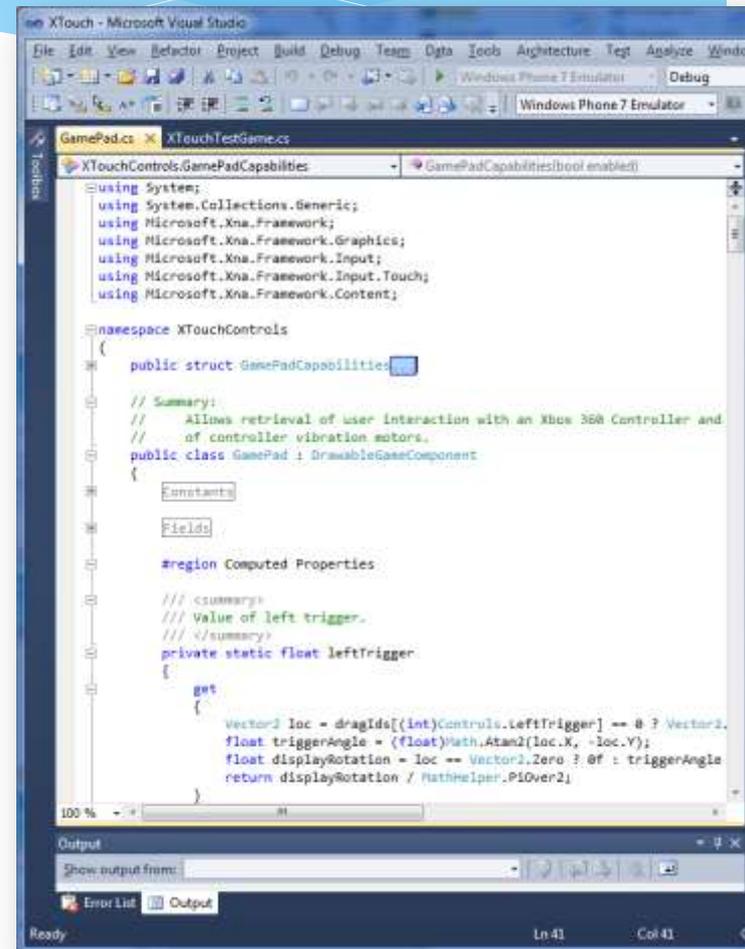


In one release we'd like to...

- \* Replace the IDE's editor (for all 50 million lines of code in all languages)
- \* to say nothing of tests 😊
- \* Replace the shell's UI and windowing system
- \* About 4,000 people involved
- \* Change the standard extensibility mechanism to MEF
- \* Millions of customers
- \* Completely rewrite the C++ project and build system
- \* Oh, you wanted to get something done as well?

# New Editor: Ideas that Worked

- \* “Prototype” by shipping
  - \* VS2010 editor shipped first in Blend
  - \* Or limit exposure (C++ projects)
- \* Old and new side-by-side during development
- \* Extensibility = componentization = testability



```
using System;
using System.Collections.Generic;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;
using Microsoft.Xna.Framework.Input;
using Microsoft.Xna.Framework.Input.Touch;
using Microsoft.Xna.Framework.Content;

namespace XTouchControls
{
    public struct GamePadCapabilities
    {
        // Summary:
        //   Allows retrieval of user interaction with an Xbox 360 Controller and
        //   of controller vibration motors.
        public class GamePad : DrawableGameComponent
        {
            Constants
            Fields

            #region Computed Properties
            /// <summary>
            ///   Value of left trigger.
            /// </summary>
            private static float leftTrigger;
            {
                get
                {
                    Vector2 loc = dragIds[(int)Controls.LeftTrigger] == 0 ? Vector2.Zero :
                    float triggerAngle = (float)Math.Atan2(loc.X, -loc.Y);
                    float displayRotation = loc == Vector2.Zero ? 0f : triggerAngle;
                    return displayRotation / MathHelper.PiOver2;
                }
            }
        }
    }
}
```

# New Editor: Ideas that Tanked

- \* “Let’s work in our own branches”
- \* “Shimming should be straightforward”
  - \* 5x bug ratio shims:core (and that’s still true today)
  - \* Mistake to let so many clients keep using shims
- \* “You just call the {native, managed} code from {managed, native}... how hard could it be?”
  - \* Undo system was single largest cause of memory and stress issues for the editor

# Lesson Productized: What Would Make this Easier?

The screenshot shows the Microsoft .NET Managed Extensibility Framework (MEF) project page on CodePlex. The page features the Microsoft .NET logo and the title "Managed Extensibility Framework". The navigation menu includes Home, Downloads, Documentation, Discussions, Issue Tracker, Source Code, People, and License. The main content area has a "Welcome to the MEF community site" section, followed by a "Downloads" section showing the current version (MEF 2 Preview 2) and its statistics (800 downloads). The "Activity" section displays page views (20656), visits (5178), and downloads (717). The "Related Projects" section lists "Base Class Libraries".

**Microsoft .NET Managed Extensibility Framework**

CodePlex Open Source Community

Search all CodePlex projects

Home Downloads Documentation Discussions Issue Tracker Source Code People License

View All Comments | Print View | Page Info | Change History (all pages)

Search Wiki & Documentation

### Welcome to the MEF community site

Application requirements change frequently and software is constantly evolving. As a result, such applications often become monolithic making it difficult to add new functionality. The Managed Extensibility Framework (MEF) is a new library in .NET Framework 4 and Silverlight 4 that addresses this problem by simplifying the design of extensible applications and components.

MEF has shipped in .NET 4 and Silverlight 4!

**Disclaimer:** This CodePlex project gives you access to the code that the MEF team is working on for future releases. The source available on this site does not necessarily reflect what will be available in future versions of .NET or Silverlight or other platform. This is, however, a great opportunity to hear from you! We welcome any feedback you have on your experience.

**Downloads**

CURRENT	MEF 2 Preview 2
DATE	Mon Oct 11 2010 at 7:00 AM
STATUS	Alpha
RATING	No Ratings
DOWNLOADS	800 downloads
MORE	<a href="#">View all downloads</a>

**Activity**

Page Views	20656
Visits	5178
Downloads	717
Application Runs	N/A

[View Detailed Stats](#)

**Related Projects**

- [Base Class Libraries](#)

**Where to go next**

[Learn more about MEF](#)  
<http://mef.codeplex.com/>

# Lessons Productized: Smaller is Better

The screenshot shows the Visual Studio Gallery website. The browser address bar displays "http://visualstudiogall...". The page header includes the Microsoft Visual Studio logo, a search bar with "Search MSDN with Bing", and the text "United States (English) Sign in". The navigation menu includes "Home", "Getting Started", "Library", "Learn", "Downloads", "Extend", "Gallery", "Support", and " Forums". Below the navigation, there are links for "Gallery Home", "My Contributions", "My Notifications", and "Feedback".

The main content area is divided into several sections:

- Visual Studio Gallery**: Products and Extensions for Visual Studio. A description states: "The Visual Studio Gallery provides quick access to tools, controls, and templates to help you get the most out of Visual Studio. You can browse, subscribe to RSS feeds and share your extensions with the Visual Studio development community. Each contribution is licensed to you under a License Agreement by its owner, not Microsoft. Microsoft does not guarantee the contribution or purport to grant rights to it."
- BROWSE**: 2,086 items in the Gallery.
- UPLOAD**: Add Products and Extensions.
- SEE MY GALLERY**: Sign in to access your Gallery.
- FIND**: Search the Visual Studio Gallery.
- POPULAR SEARCHES**: Visual Studio 2010, Microsoft, power commands, T4, power tools, theme, Coding.

The main content area is divided into three columns:

- RECENTLY ADDED**:
  - Feature Builder Power Tools** (Free) by Visual Studio Architecture Power Tools (Microsoft). Description: "Feature Builder is a Power Tool for Visual Studio 2010 which helps you easily create rich Visual Studio extensions. You can quickly..."
  - RADvolution Designer - Database Edition 2010v2** (Trial) by DevelopGuidance. Description: "Rapidly build professional SQL database..."
- MOST POPULAR**:
  - Productivity Power Tools** (Free) by Microsoft. Description: "A set of extensions to Visual Studio Professional (and above) which improves developer productivity."
  - Power Commands for Visual Studio 2010** (Free) by Microsoft.
- HIGHEST RATED**:
  - H3Viewer - Help Viewer for VS 2010** (Free) by Rob Chandler [Help MVP]. Description: "Replacement offline help viewer for VS 2010. Similar to the old VS 2008 DExplore."
  - Sitecore Rocks** (Free) by JakobChristensen (Sitecore). Description: "Sitecore integration with Microsoft Visual..."

A yellow oval highlights the "Productivity Power Tools" listing in the "MOST POPULAR" section.

# Lesson Learned: Agile + Portfolio Management



# Shorter is Better

Dev11

TFS 2010

TFS 2008

October

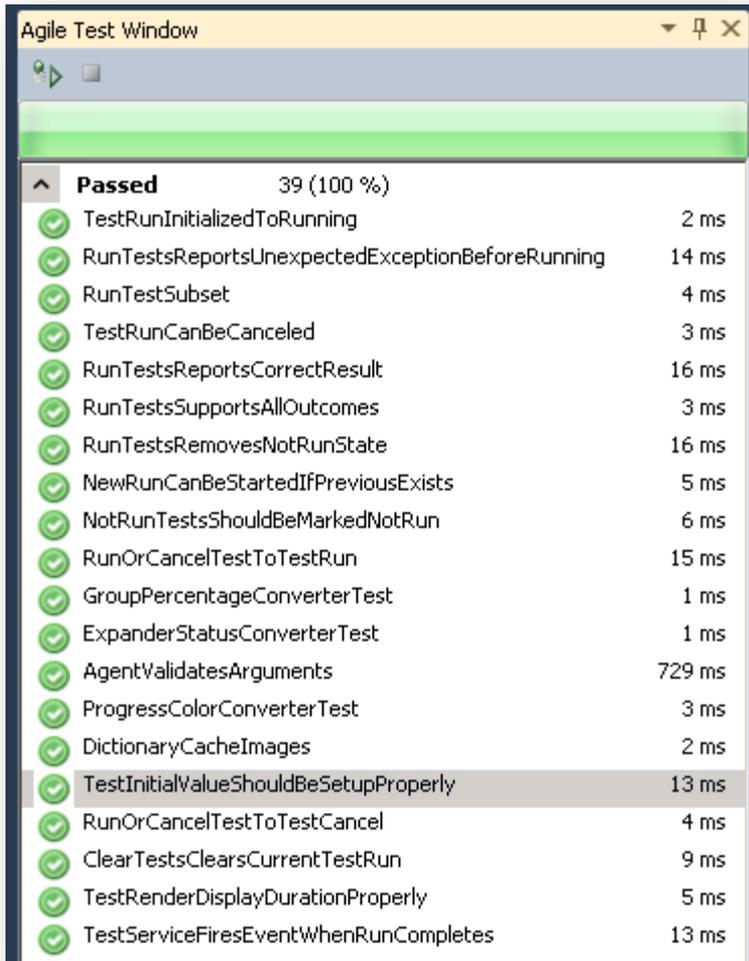
November

December





# Lessons Productized: Double Down on Agile



Agile Test Window

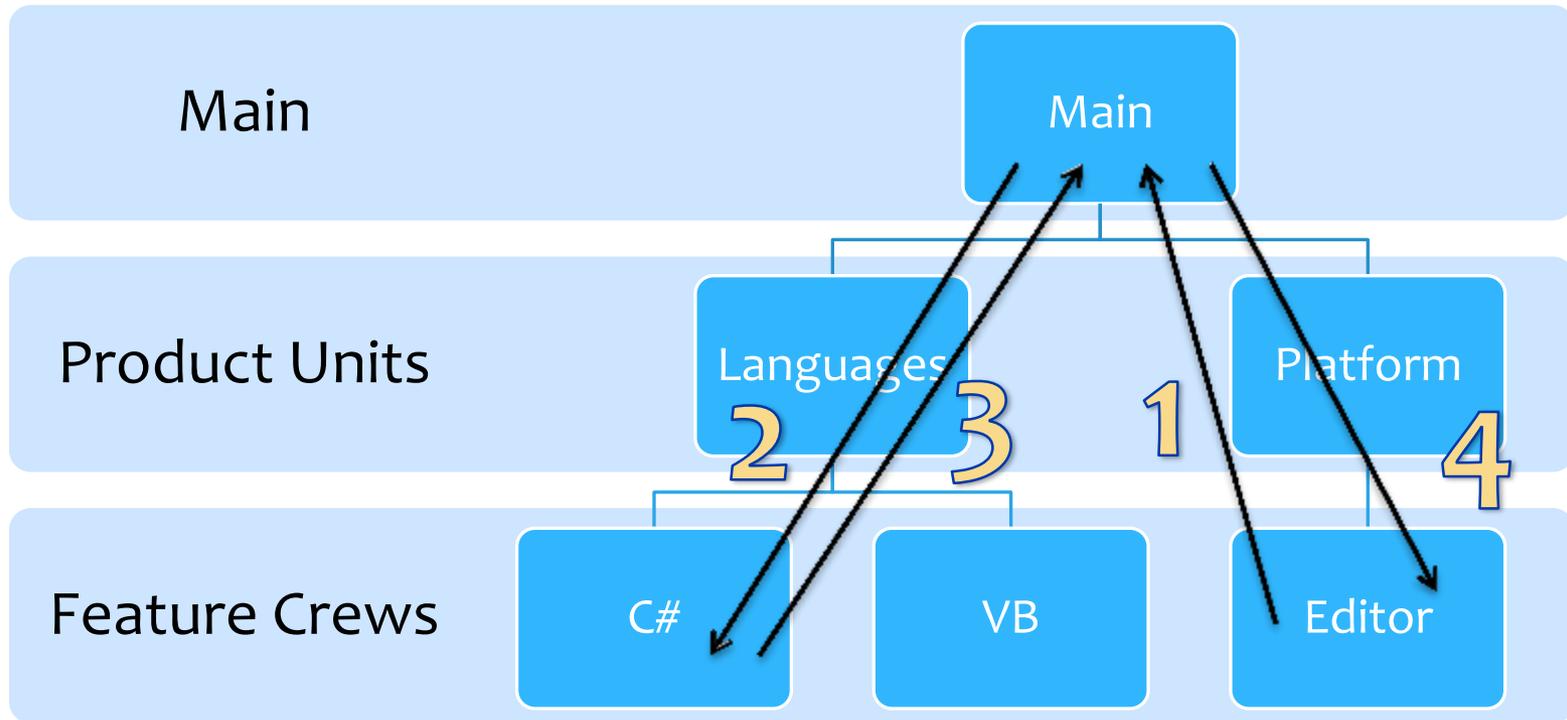
Passed 39 (100 %)

TestRunInitializedToRunning	2 ms
RunTestsReportsUnexpectedExceptionBeforeRunning	14 ms
RunTestSubset	4 ms
TestRunCanBeCanceled	3 ms
RunTestsReportsCorrectResult	16 ms
RunTestsSupportsAllOutcomes	3 ms
RunTestsRemovesNotRunState	16 ms
NewRunCanBeStartedIfPreviousExists	5 ms
NotRunTestsShouldBeMarkedNotRun	6 ms
RunOrCancelTestToTestRun	15 ms
GroupPercentageConverterTest	1 ms
ExpanderStatusConverterTest	1 ms
AgentValidatesArguments	729 ms
ProgressColorConverterTest	3 ms
DictionaryCacheImages	2 ms
TestInitialValueShouldBeSetupProperly	13 ms
RunOrCancelTestToTestCancel	4 ms
ClearTestsClearsCurrentTestRun	9 ms
TestRenderDisplayDurationProperly	5 ms
TestServiceFiresEventWhenRunCompletes	13 ms

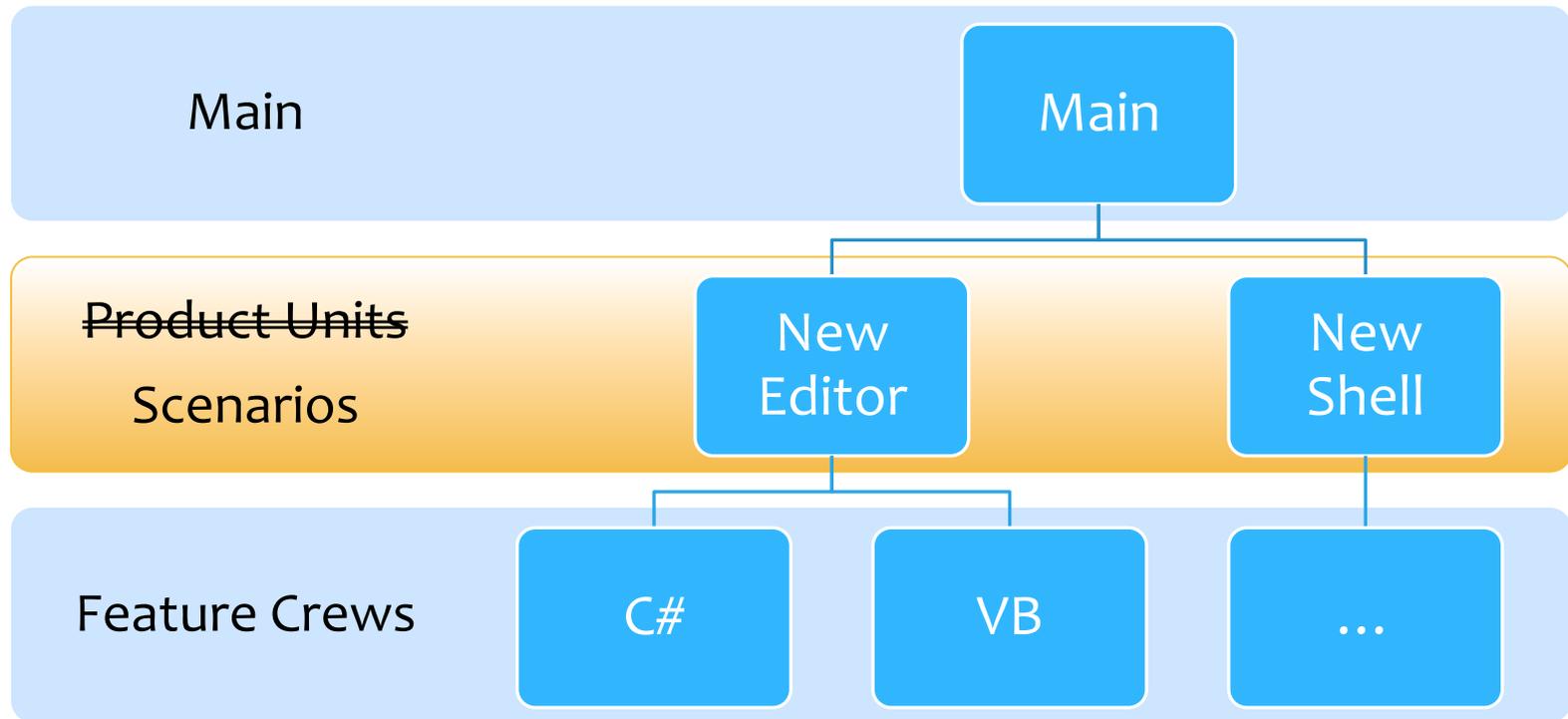
## Research Trends

- \* Unit test discovery and path analysis
- \* Detect code “repeats” and suggest fixes
- \* Mocking frameworks and techniques
- \* Statistical analysis of bugs and bug fixes

# Branching Mistakes



# Branching Mistakes



# Internal Code Motion Dashboards

Main

Main  
Build 34

Level 1

Team A, build 22  
4 Tests failing  
Last FI: 510/1  
Last RI: 10/10

Team B, build 30  
All tests passing  
Last FI: 10/20  
Last RI: 10/18

Level 2

...

...

...

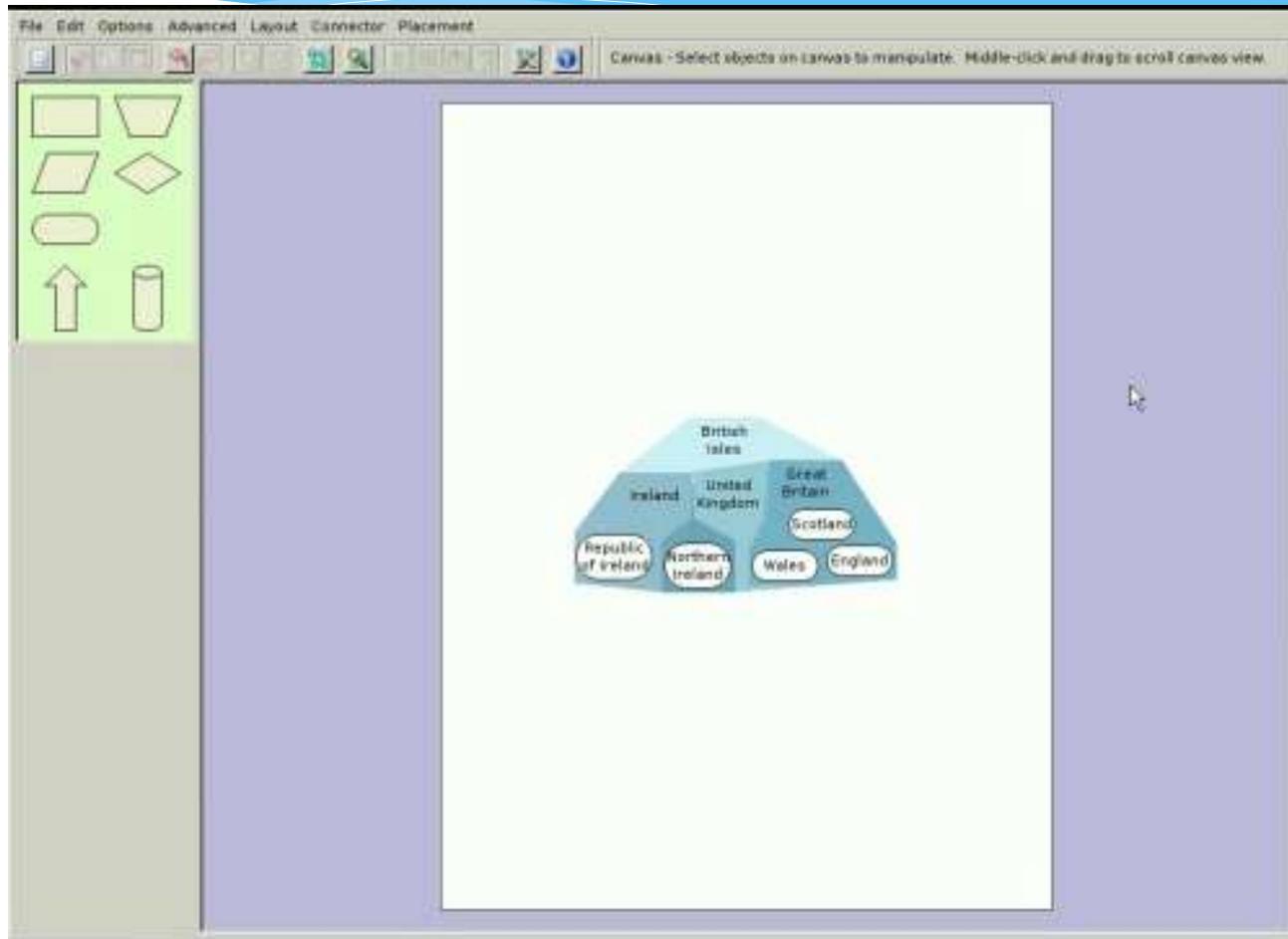
# Untangling Spaghetti



# Spaghetti Demo - Takeaways

- \* Assembly-level analysis for large “brown fields”
- \* Tolerance for legacy mistakes and business needs
  - \* <permit>dependency we don't like</permit>
- \* Usability at scale
  - \* World view
  - \* Flexible, incremental layout engine
  - \* “Semantic zoom” to present most relevant information at all zooming levels (just like mapping software)

# When Usability *is* Functionality



# Where are my Reading Glasses?



# Microsoft Visual Studio 2010 Ultimate

- Connect To Team Foundation Server
- New Project...
- Open Project...

## Recent Projects

- WindowsFormsApplication1
- CreateManagedInstance
- IsMetadataDispenserFreeThreaded
- CoRegisterMessageFilterTest

## Get Started

### Guidance and Resources

### Latest News

- Welcome
- Windows
- Web
- Cloud
- Office
- SharePoint
- Data



### What's New in Visual Studio 2010

Learn about the new features included in this release.

- [Visual Studio 2010 Overview](#)
- [What's New in the .NET Framework version 4](#)
- [Customize the Visual Studio Start Page](#)



### Creating Applications with Visual Studio



### Extending Visual Studio



### Community and Learning Resources

Close page after project load

Show page on startup

# Shell Renovation Plan: Staged Refactoring

- \* “Reverse engineer” a spec
- \* Find or write characterization tests
- \* Define the data models
- \* Replace the main window with WPF
- \* Write new...
  - \* Window Manager, Command Bar presentation
  - \* Hidden behind switches, off by default
- \* Scout with selected teams
- \* Test functionality, perf, stress, e2e, memory, remote, VM, ...
- \* Reverse the switches
  - \* Leave old presentation for regression testing
- \* Remove old code (and ship 😊).

# What Could Go Wrong?

- \* A lot of things that we anticipated...
  - \* Code that relied on HWNDs (estimated about right)
  - \* Tests that relied on HWNDs
    - \* Underestimated size and scope of problem, including the diversity of these tests
  - \* Significant cross-divisional functionality testing
- \* And then some we didn't...
  - \* Significant responsiveness issues (retread, interop)
    - \* Responsiveness is suddenly part of characterization tests!
    - \* Menu drop...
  - \* Customer headaches...literal ones!

# Lessons Learned: Display Modes

**Ideal**      The quick brown fox jumps over the lazy dog

**Display**    The quick brown fox jumps over the lazy dog

**Ideal**      `i = i+1; j++; // ++++++++`

**Display**    `i = i+1; j++; // ++++++++`

# Lessons Learned: Display Modes

\* Ideal

オプションおよびプロパティ

\* Display

オプションおよびプロパティ

# Lessons Productized

- \* Offer display mode, fix gamma settings
  - \* *Pick a familiar default – you can't force customers into happiness!*
  - \* Test (literally) for pixel-parity; anything less is subject to interpretation
- \* Diagnostics to capture and understand IDE “in the wild”
  - \* Video driver nightmares
- \* Responsiveness tracking
  - \* Preserving remote desktop optimization
- \* Identify anti-patterns... educate for now, consider “fingerprinting” later

# Feedback, Detection, and Diagnosis

**Single biggest challenge: Issues we can't diagnose in house**

- \* Functionality – Watson
- \* Responsiveness – PerfWatson
- \* Dogfooding feedback – VS “send a smile” tool
- \* In-the-wild problems (video drivers)
  - \* Built-in tools: Help → About → dxdiag
  - \* Opt-in tools: SQM
  - \* “on demand” tools: Mostly perf analyzers today

# Dirty Laundry



# VS 2010 Customer Survey

Count	Performance Issue
193	Overall slowness
168	Startup takes too long
139	Intermittent slowdowns

# Software Components

## They're awesome!

- \* Dynamically composable and extensible
- \* Decoupled services, teams, and delivery dates
- \* GC will solve all problems 😊
- \* Independently testable

## They're terrible!

- \* Unpredictable once combined
- \* Emergent performance and stress problems
  - \* Leaks, responsiveness, ...
- \* End-to-end customer testing is the only source of truth

# Lessons Productized: PerfWatson (aka “no more spinner”)

#Hits	Hit%	Total Delay(s)	Delay%	Avg Delay	Name
4222	100%	25,027	100%	5	Root
4222	100%	25,027	100%	5	devenv ( 999)
4222	100%	25,027	100%	5	tid ( 100)
1284	30%	14,487	57%	11	ntdll!_RtlUserThreadStart
1283	30%	14,485	57%	11	ntdll!_RtlUserThreadStart
1283	30%	14,485	57%	11 *	kernel32!BaseThreadInitThunk
530	12%	1,730	6%	3	devenv!__tmainCRTStartup
530	12%	1,730	6%	3	devenv!WinMain
530	12%	1,730	6%	3	devenv!CDevEnvAppId::Run
530	12%	1,730	6%	3 *	=> devenv!util_CallVsMain
504	11%	1,637	6%	3	=> msenv!VStudioMain
504	11%	1,637	6%	3	=> msenv!VStudioMainLogged
504	11%	1,637	6%	3	=> msenv!CMsoComponent::PushMsgLoop
504	11%	1,637	6%	3	=> msenv!SCM_MsoCompMgr::FPushMessageLoop
504	11%	1,637	6%	3	=> msenv!SCM::FPushMessageLoop
504	11%	1,637	6%	3	=> msenv!CMsoCMHandler::FPushMessageLoop
504	11%	1,637	6%	3	=> msenv!CMsoCMHandler::EnvironmentMsgLoop
504	11%	1,637	6%	3	=> msenv!SCM_MsoStdCompMgr::FDoIdle
504	11%	1,637	6%	3	=> msenv!SCM::FDoIdle
504	11%	1,637	6%	3	=> msenv!SCM::FDoIdleLoop
380	9%	1,265	5%	3	csproj!CLangPackage::FDoIdle
380	9%	1,265	5%	3	csproj!CVsProject::FDoIdle
380	9%	1,265	5%	3	csproj!CVsProject::InitF5HostingProcess

# Lessons Productized: PerfWatson (aka “no more spinner”)

- \* UI hangs (“spinner”) triggers PerfWatson
- \* Snapshot of stack is taking and sent to server
- \* Server aggregates traces...
  - \* The greater the delay and the more reports of that trace, the higher it rises in the ranking
- \* Provides a prioritized, pre-diagnosed list of places to go improve responsiveness
- \* Naturally aggregates across all components

# Lessons Learned: Memory is Finite

VMMMap - Sysinternals: www.sysinternals.com

File Edit View Options Help

Process: devenv.exe  
PID: 4320

Committed: 592,516 K

Private Bytes: 281,280 K

Working Set: 325,344 K

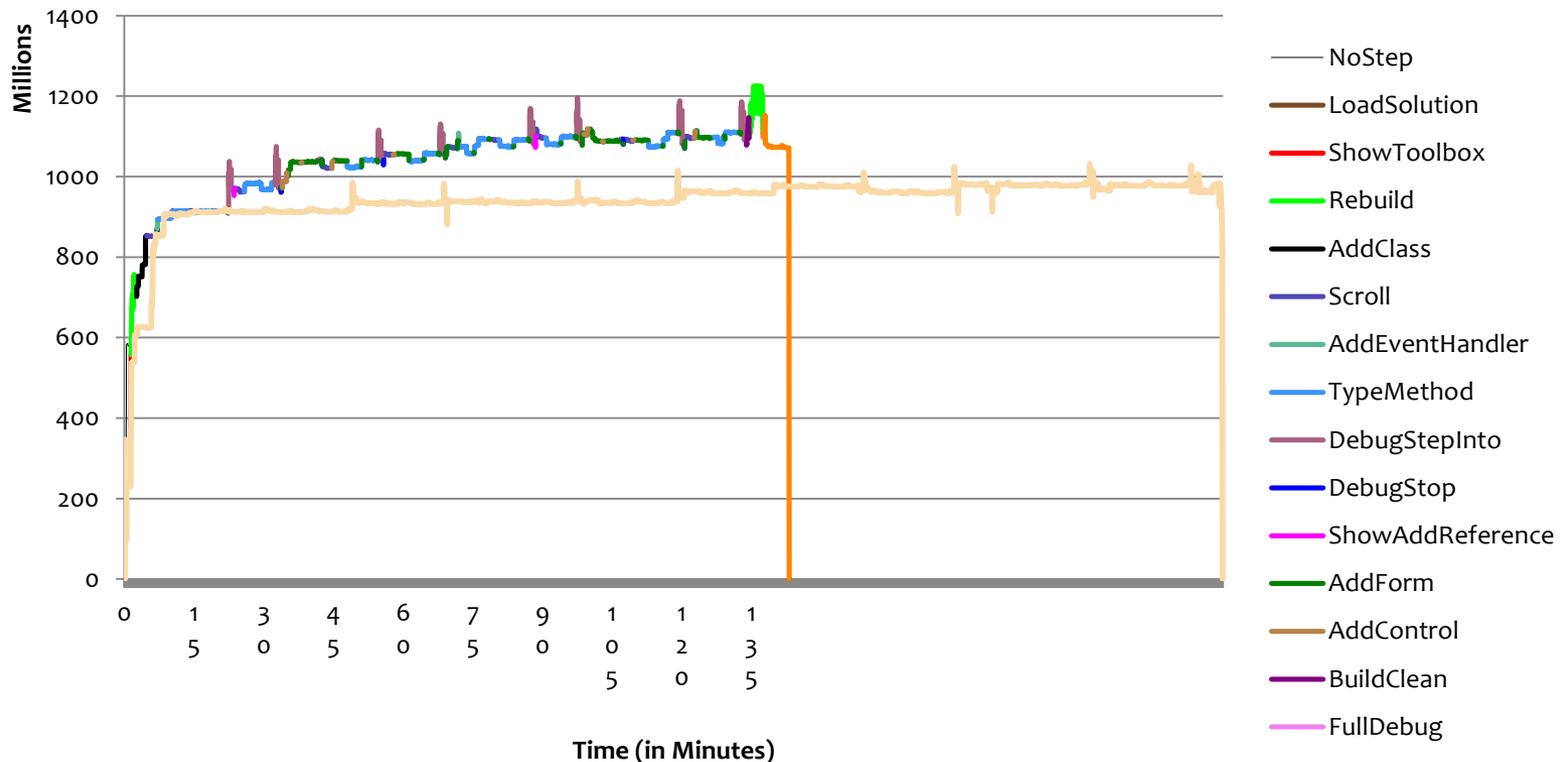
Type	Size	Committed	Private	Total WS	Private WS	Shareable WS	Shared WS	Locked WS	Blocks	Largest
Total	705,628 K	592,516 K	281,280 K	325,344 K	232,704 K	92,640 K	36,036 K		3166	
Image	370,664 K	369,988 K	130,840 K	195,120 K	115,560 K	79,540 K	31,380 K		2384	24,872 K
Mapped File	56,332 K	56,332 K	9,956 K	9,956 K	9,956 K	9,956 K	1,928 K		48	19,796 K
Shareable	48,452 K	15,692 K	3,136 K	3,136 K	3,136 K	3,136 K	2,720 K		62	12,288 K
Heap	75,000 K	39,892 K	39,828 K	39,084 K	39,080 K	4 K	4 K		131	16,192 K
Managed Heap	70,336 K	45,764 K	45,764 K	45,636 K	45,636 K				111	32,768 K
Stack	35,200 K	22,528 K	22,528 K	716 K	716 K				184	1,024 K
Private Data	45,424 K	38,100 K	38,100 K	27,476 K	27,472 K	4 K	4 K		246	9,344 K
Page Table	4,220 K	4,220 K	4,220 K	4,220 K	4,220 K					
Free	1,395,680 K									249,536 K

Address	Type	Size	Committed	Private	Total WS	Private ...	Sharea...	Share...	Lock...	Blocks	Protection	Details
01E50000	Image	24,872 K	24,872 K	22,316 K	23,644 K	22,120 K	1,524 K			10	Execute/Read	C:\Program Files\Microsoft Visual Studio 11.0\Common7\ND...
56710000	Image	17,332 K	17,332 K	2,368 K	5,664 K	396 K	5,268 K	4,332 K		41	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Prese...
07470000	Image	14,156 K	14,156 K	9,032 K	10,412 K	6,944 K	3,468 K	1,884 K		20	Execute/Read/Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\msco...
164C0000	Image	13,504 K	13,504 K	12,304 K	12,848 K	11,940 K	908 K			16	Execute/Read	C:\Program Files\Microsoft Visual Studio 11.0\Common7\ND...
7B2A0000	Image	12,584 K	12,584 K	1,816 K	3,292 K	204 K	3,088 K	1,968 K		35	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Syste...
76820000	Image	12,580 K	12,580 K	52 K	780 K	24 K	756 K	756 K		5	Execute/Read	C:\Windows\System32\shell32.dll
549B0000	Image	10,976 K	10,976 K	1,604 K	4,592 K	260 K	4,332 K	2,400 K		31	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Prese...
648A0000	Image	10,560 K	10,560 K	1,152 K	340 K	60 K	280 K			18	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Syste...
126A0000	Image	10,488 K	10,488 K	8,904 K	9,388 K	8,640 K	748 K			12	Execute/Read	C:\Program Files\Microsoft Visual Studio 11.0\VB\Bin\msvb...
331E0000	Image	9,108 K	9,108 K	1,216 K	1,068 K	128 K	940 K			32	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Micro...
7A5F0000	Image	8,696 K	8,696 K	1,408 K	4,036 K	224 K	3,812 K	2,524 K		31	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Syste...
20B20000	Image	7,640 K	7,640 K	7,024 K	7,208 K	6,780 K	428 K			16	Execute/Read	C:\Program Files\Microsoft Visual Studio 11.0\Common7\ND...
17B70000	Image	7,056 K	7,056 K	4,056 K	5,216 K	3,104 K	2,112 K			16	Execute/Read/Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Micro...
088E0000	Image	6,828 K	6,828 K	5,940 K	5,148 K	4,316 K	832 K	600 K		12	Execute/Read/Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Syste...
1C3D0000	Image	6,592 K	6,592 K	3,468 K	3,268 K	2,788 K	480 K	76 K		12	Execute/Read/Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Syste...
6B3C0000	Image	6,472 K	6,472 K	316 K	3,532 K	84 K	3,448 K	2,520 K		16	Execute/Read/Write	C:\Windows\Microsoft.NET\Framework\v4.0.30319\clr.dll
68C70000	Image	5,788 K	5,788 K	544 K	1,200 K	68 K	1,132 K	1,012 K		13	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Syste...
108A0000	Image	5,152 K	5,152 K	3,556 K	3,856 K	2,736 K	1,120 K			14	Execute/Read/Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Micro...
31200000	Image	5,136 K	5,136 K	904 K	944 K	132 K	812 K			25	Execute/Copy on Write	C:\Windows\assembly\NativeImages_v4.0.30319_32\Micro...

Timeline... Heap Allocations... Call Tree... Trace...

# Memory Analysis Over Time ("Stress" and end-to-end runs)

VirtualBytes:Picasso Short Haul E2E (Dev10).1627824.1  
Ultimate + Windows 7, vs\_langs 21214.00 High-End



# 'Debugging' Memory

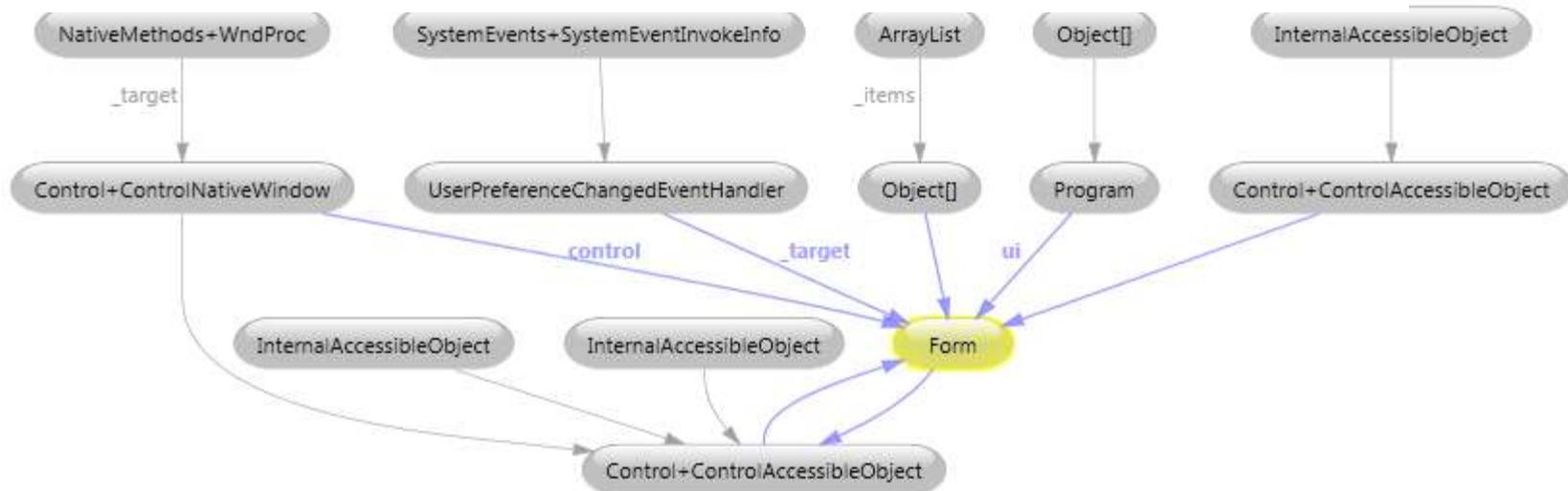
## SOS.dll (SOS Debugging Extension)



.NET Framework 4 | Other Versions ▾

The SOS Debugging Extension (SOS.dll) helps you debug managed programs in the WinDbg.exe debugger and in Visual Studio by providing information about the internal common language runtime (CLR) environment.

The SOS Debugging Extension is automatically installed with Visual Studio and requires your project to have unmanaged debugging enabled. You can also use the SOS Debugging Extension with the Windows debugger (WinDbg.exe).



# Memory Profiler and Managed Leak Analysis

\* F1 Demo

# Lessons Learned

- \* Managed code leaks...
  - \* GC is great for preventing errors, but leaks are hard to find without memory regression analysis tools
- \* ...but interop'ed code spews
  - \* Collision of different memory management strategies (COM, native to managed/GC)
  - \* Need tools and training to isolate “boundary” problems
- \* Perf testing improvements...

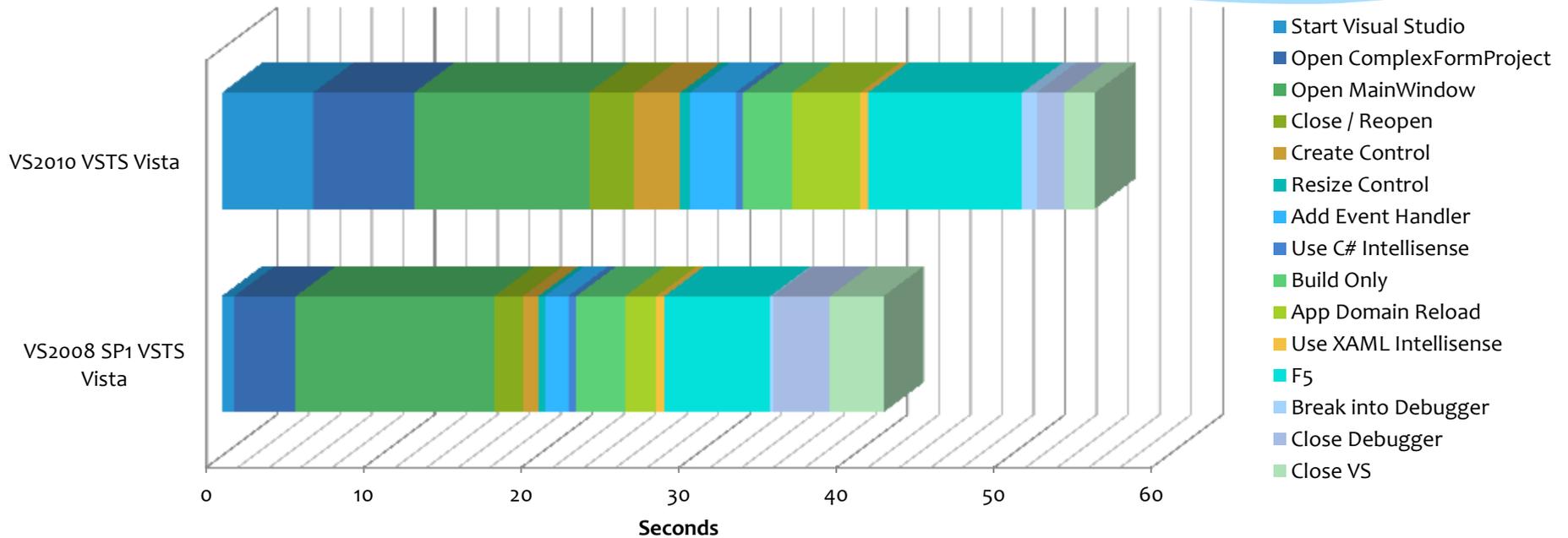
# A Changing View of Perf Testing

**Reality check: The test matrix is infinite.**

- \* In house automation → Better in-the-wild diagnostics
- \* Time perf → Responsiveness analysis
- \* Regression analysis → Scenario/OGF focus
- \* Repeatability → Heterogeneity (VMs, remote, ...)
  - \* If you turn off virus checkers, what happens if that's the bug?
- \* Internal examples → Real customer solutions
- \* Microbenchmarks → Multi-step end-to-ends
- \* Rollups of deltas → Customer scorecards/gaps

# C# WPF XAML

Cider 20305.20306



# Blend Dogfooding OGF – Large C# Solution [AndreHal]

## Resolved Issues (no longer in flight)

Expected OGF: Good

Current OGF: Fair

Build: 21216 (Main)

Gap to Goal: 1 OGF Level (11 Bugs)

### OGF Impacting Fixes

Description	Bug ID	Owner	PU	Fixed In	In Main	Comments
<b>Fixed in Main 1204 (current dogfood build)</b>						
Cannot hit all breakpoints in the Expression Blend solution	823959/788188	Michael Lehenbauer	VSP	10/15 VSP	Y	
ALIGN 16 for an asm constant is not ending up aligned in the image	819251	Vance Morrison	CLR	11/16 Tools 11/23 RC1Rel	Y	
VS is leaking GDI handles during debugging.	824214	Jim Griesmer	TeamEng	11/9 lab26vsts	Y	
<b>Fixed in Main 1216 (next dogfood build)</b>						
Edit and continue functionality is broken in the Expression Blend solution	824918	Barry Nolte	TeamEng	12/3 lab26vsts	Y	ENC not working is by design due to the assembly being App-Domain Neutral [workaround in place]. Debugger checked in an improved error message to clarify the reason.
Random error dialogs pop up and crashes when editing Blend XAML files inside VS	824167	Kevin Pilch-Bisson	VS Langs	12/7 vs_langs0	Y	
Crash on opening XAML / using intellisense inside the Blend solution	829302	Eric Fisk	WPF	12/7 vs_langs	Y	
Crash after typing some text in XAML using the Blend solution using xaml async mode	829988	Eric Fisk	WPF	12/7 vs_langs	Y	
Editor may become blocked for a long time shortly after a solution is opened	829940	Dmitry Goncharenko	VSL	12/15 vs_langs	Y	

### Resolved OGF impacting “not fixed”

Description	Bug ID	Owner	PU	Resolution	Resolved Date	Comments
Conditional breakpoints are slower with CLR v4	829295	Closed	CLR	Won't Fix	12/5	Result of a CLR 4.0 architectural change. Corner case scenario in the Blend solution where BP is in an event handler fired frequently, and condition triggers 3 func-evals
Work with documents gets really sluggish and CPU pegs at 50% after making a large XAML file dirty	824154	Closed	Cider	Not Repro		Issue no longer repros in current builds
Potential perf improvement to managed stepping by reducing UTF8 to Unicode conversion in CCompilerandTrav::next	834153	Closed	VC	By Design	12/11	Cannot fix because this is the way the symbol system was design to work for glob/loc reasons

# Wrapup - Themes

- \* Scaling up isn't just size... it's population diversity
- \* Manage feature portfolios intelligently
  - \* Big rock(s) *and* agile development, not “or”
- \* Customer feedback trumps your “rational” decisions
- \* Hippocratic Oath for architecture (trust but verify)
- \* Test componentized systems for emergent problems

# Q&A, links

- \* Learn more about Visual Studio:  
[www.visualstudio.com](http://www.visualstudio.com)
- \* See components and extensions in the VS Gallery:  
[www.visualstudiogallery.com](http://www.visualstudiogallery.com)
- \* Hear about VS development processes and TFS on Brian Harry's blog: [blogs.msdn.com/bharry](http://blogs.msdn.com/bharry)

# From Lessons Learned to Lessons Productized

Dr. Tim Wagner

Visual Studio Director of Development

QCon 2010, SF