Microsoft

Visual Studio

Team Foundation Server 2013

Jason Chan
Developer Evangelist
Microsoft

Agenda

- TFS Overview
- Hardware & OS Requirements
- Application Lifecycle Management
  - **Project Management**
    - Work Item Management
    - Kankan Board view, Team Favorites, Email trigger
    - Team Project Portal
  - **Development**
    - Code Quality Tools – Storyboarding, Code Review, Code Lens, Code Analysis
    - Source Control Management
    - Feedback Client
  - **Reporting**
    - Overview
    - Built in reports

Server OS Requirements

- 64-bit versions of Windows Server 2008 R2 [1] with SP1 (Standard, Enterprise or Datacenter editions)

- Windows Small Business Server 2011 [1] with SP1 (Standard, Essentials, or Premium Add-on editions)

- 64-bit versions of Windows Server 2012 (Essentials, Standard or Datacenter editions)

- 64-bit RTM version of Windows Server 2012 R2 (Essentials, Standard or Datacenter editions)

http://msdn.microsoft.com/en-us/library/dd578592.aspx

SQL Server Requirements

Regardless of which option you choose, your instance of SQL Server must meet the following requirements:

Supported editions	• The next version of SQL Server (Express,[1] [2] Standard,[1] and Enterprise editions) • SQL Server 2012 with SP1[3] (Express,[1] [2] Standard,[1] or Enterprise Editions)
Required for Team Foundation Server	• Database Engine Services • Full-Text and Semantic Extractions for Search
Required for reporting	• Reporting Services – Native • Analysis Services
Collation settings	• Must be accent sensitive • Must not be case sensitive • Must not be Binary • Must not be Binary - code point For more information, see SQL Server Collation Requirements for Team Foundation Server.
Authentication	Windows authentication
Service account	You can use a domain account or a built-in account.

http://msdn.microsoft.com/en-us/library/dd631889.aspx

TFS 2013 Hardware Recommendation

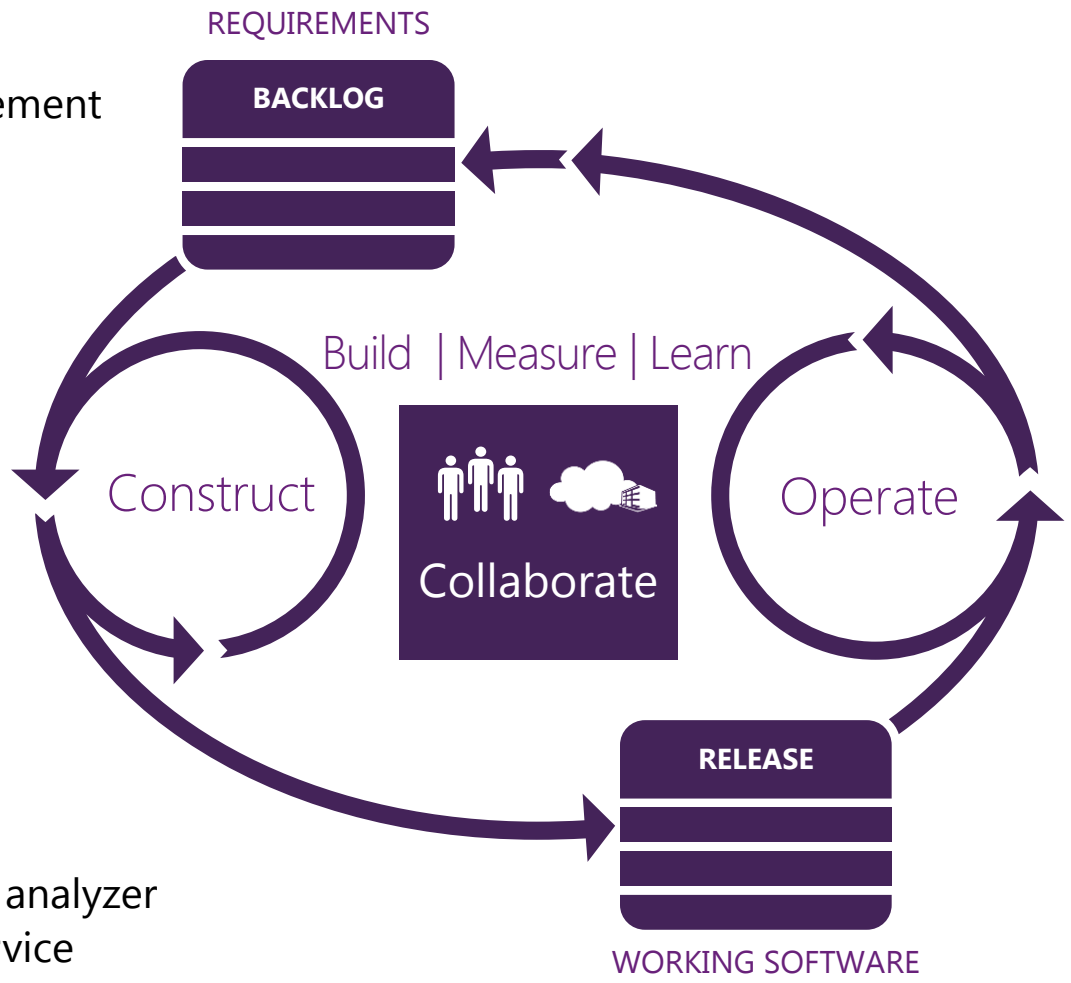
Number of users	Configuration	CPU	Memory	Hard disk
Fewer than 250 users	Single-server (Team Foundation Server and the Database Engine on the same server).	1 single core processor at 2.13 GHz	2 GB	1 disk at 7.2k rpm (125 GB)
250 to 500 users	Single-server.	1 dual core processor at 2.13 GHz	4 GB	1 disk at 10k rpm (300 GB)
500 to 2,200 users	Dual-server (Team Foundation Server and the Database Engine on different servers). This row is for Team Foundation Server.	1 dual core Intel Xeon processor at 2.13 GHz	4 GB	1 disk at 7.2k rpm (500 GB)
	This row is for the Database Engine with 500 to 2,200 users.	1 quad core Intel Xeon processor at 2.33 GHz	8 GB	SAS disk array at 10k rpm (2 TB)
2,200 to 3,600 users	Dual-server. This row is for Team Foundation Server.	1 quad core Intel Xeon processor at 2.13 GHz	8 GB	1 disk at 7.2k rpm (500 GB)
	This row is for the Database Engine with 2,200 to 3,600 users.	2 quad core Intel Xeon processors at 2.33 GHz	16 GB	SAS disk array at 10k rpm (3 TB)

http://msdn.microsoft.com/en-us/library/dd578592.aspx

Team Foundation Server 2013

Plan

- Agile Project Management
- Kanban Board
- Work item tagging
- Work item charting



Operate

- Visual Studio and System Center integration
- Performance events

Develop

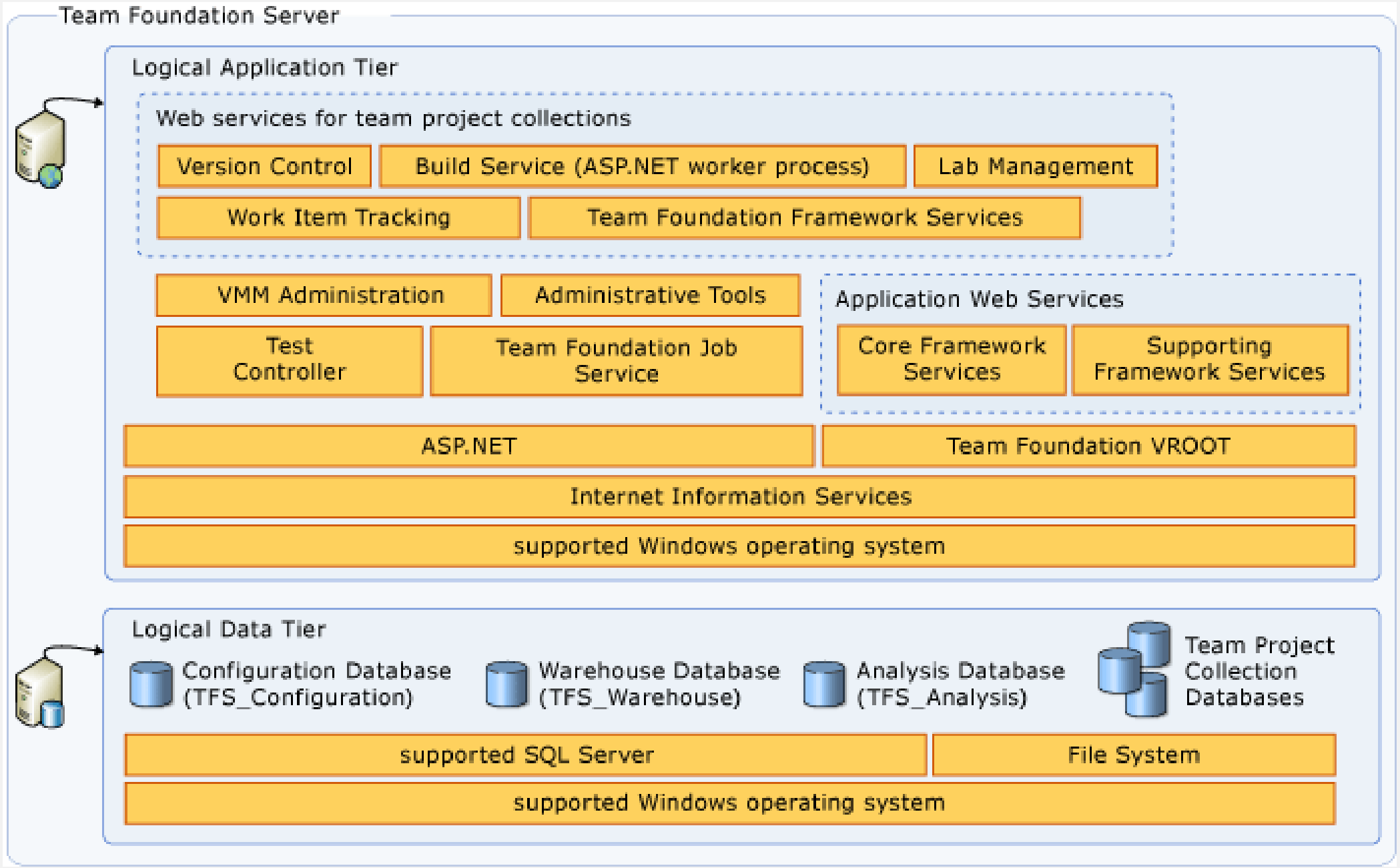
- Team Room
- Git
- CodeLens
- .NET memory dump analyzer
- Load testing as a service

Release

- Integrated release management
- Configuration-based deployments

Team foundation server 2013

- Tightly integrated with Visual Studio 2013
- Built on:
  - Windows SharePoint Foundation
  - SQL Server 2008/2012
  - SQL Server Reporting Services, Analysis Services



Visual Studio

TFS Databases

Database Name and Description
TFS_Configuration This database stores the catalog of resources and the configuration information for Team Foundation Server. This database contains the operational stores for Team Foundation Server.
TFS_Warehouse This database stores the data for reports.
TFS_Analysis This multi-dimensional database stores the aggregated data from team project collections.
Databases for team project collections One database for each team project collection, containing data from all team projects in that collection.

Team Foundation Server 2013

- Full suite of integrated ALM capabilities
  - Work Item Tracking
  - Source Control
  - Feedback Support
  - Code Review
  - Collaboration/Knowledge Sharing
  - Project Tracking/Reporting
  - Project Dashboards
  - Build Management
  - Continuous Integration
  - Test Lab Management
  - Team Room
  - Code Lens

Conflicting Priorities

Changing requirements and shifting priorities

Poor communication, long delivery cycles, rework, context switching, and customer dissatisfaction

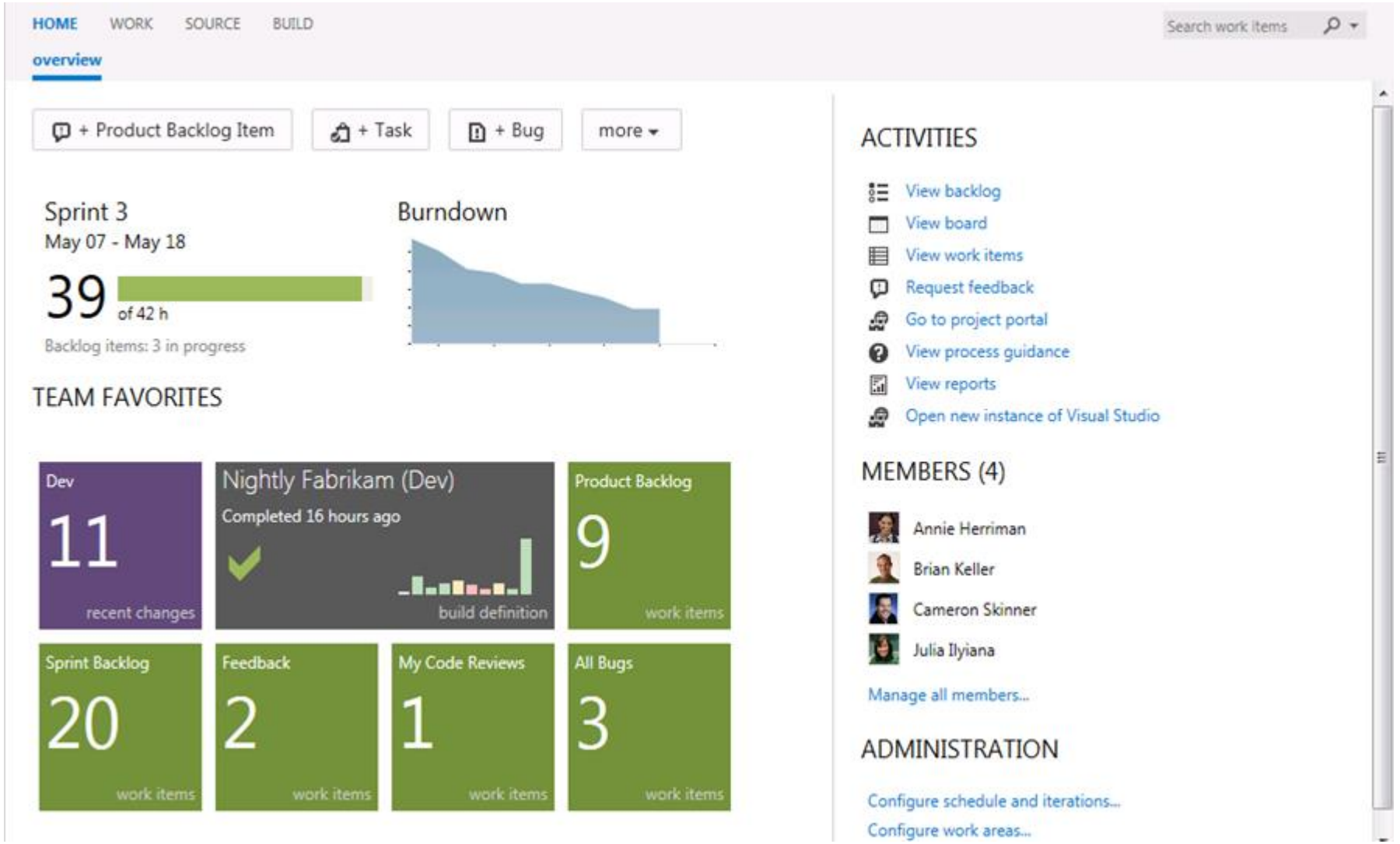
Backlog and task management tools

Streamlined tool for creating, prioritizing and managing user requirements and tasks

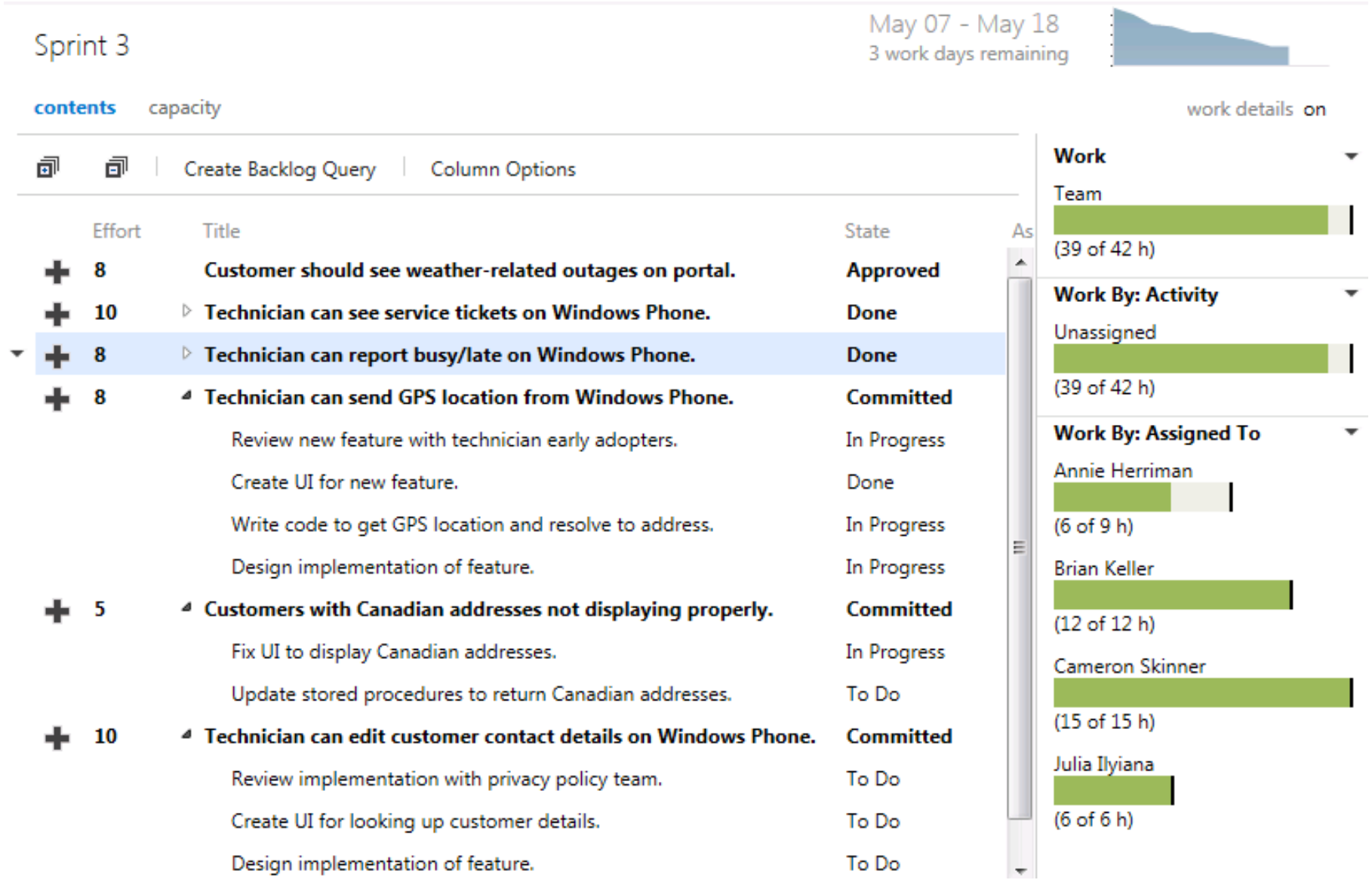
Demonstration & Labs

- Team Web Access
- Working with workitems
- Planning your Projects with team foundation server 2013

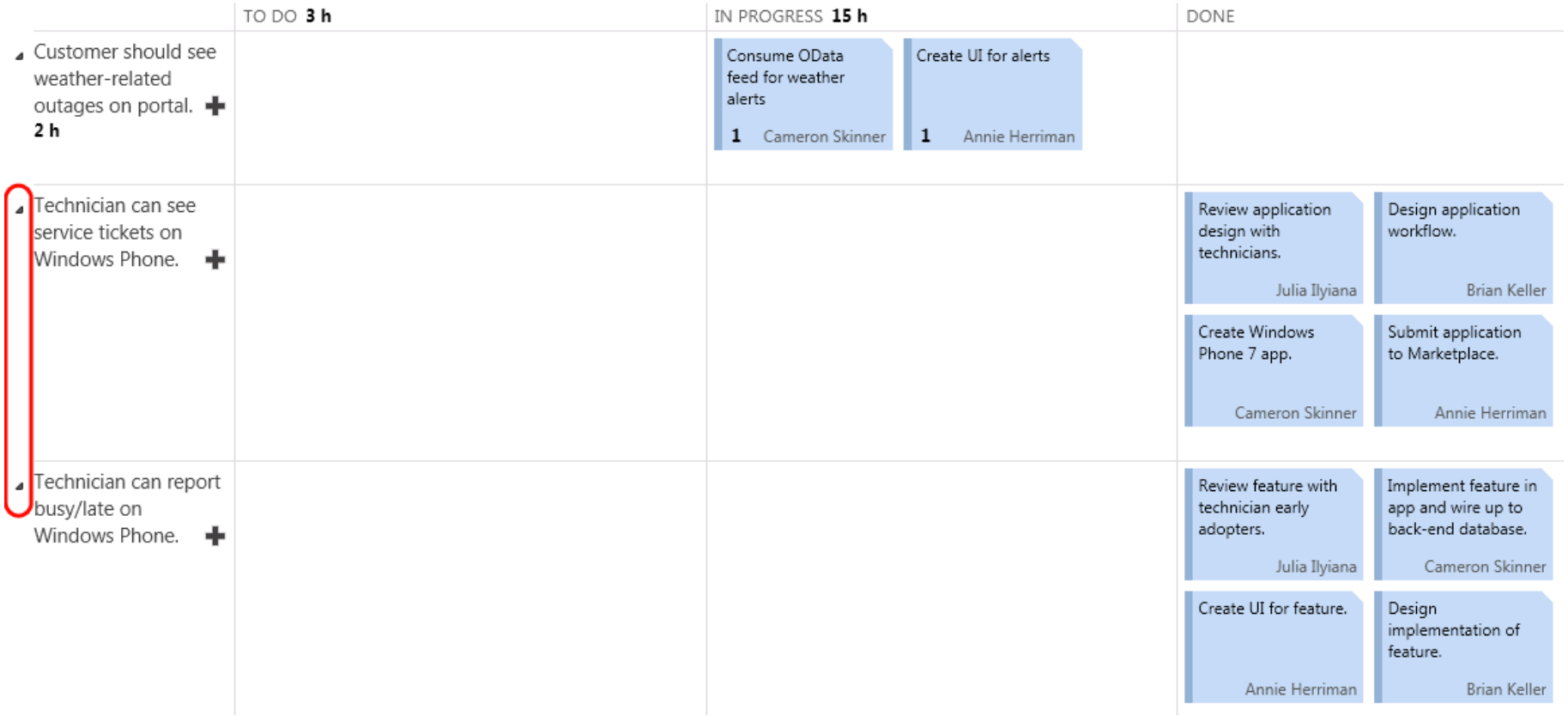
Team Web Access (TFS)



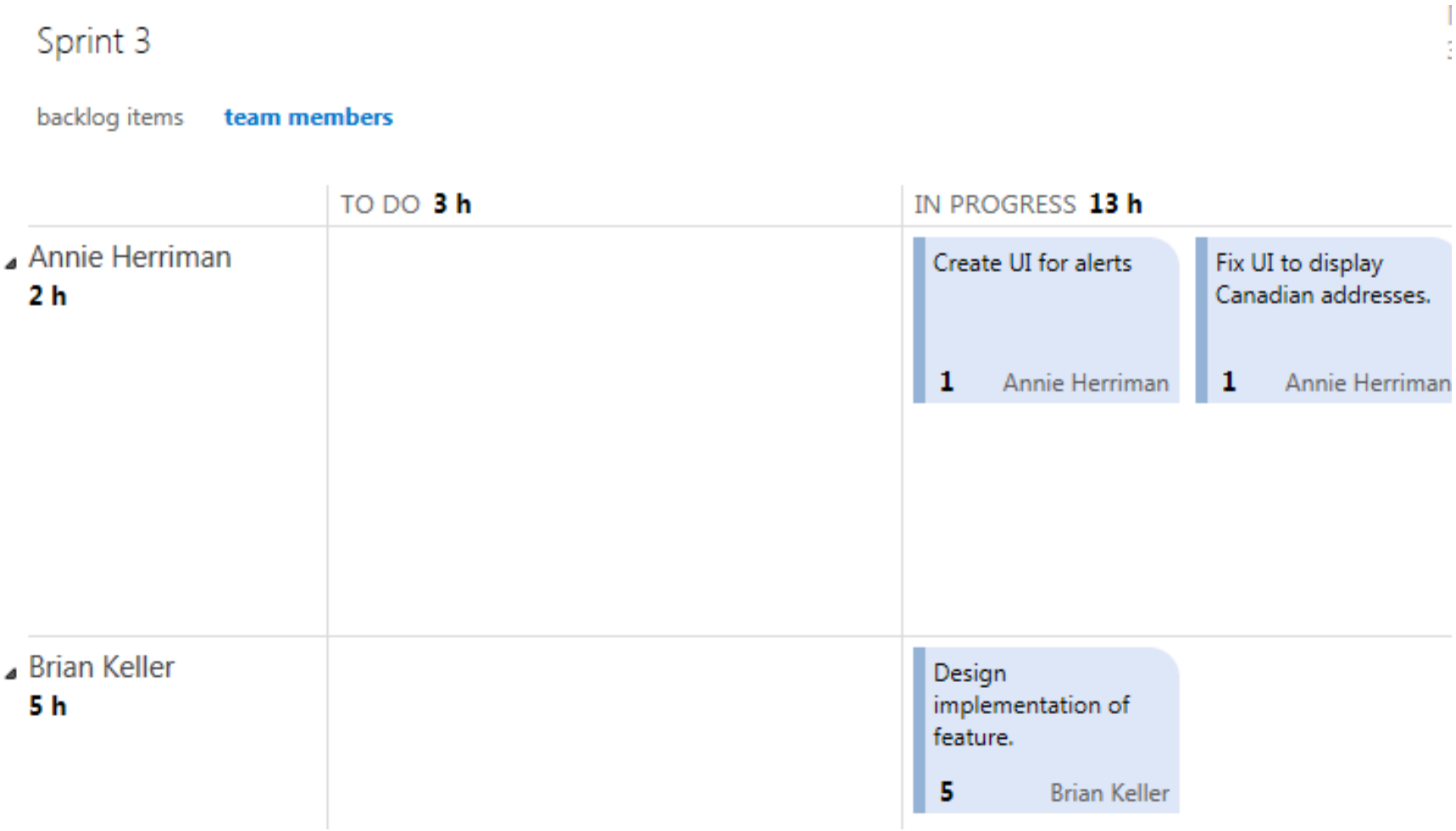
Project Management



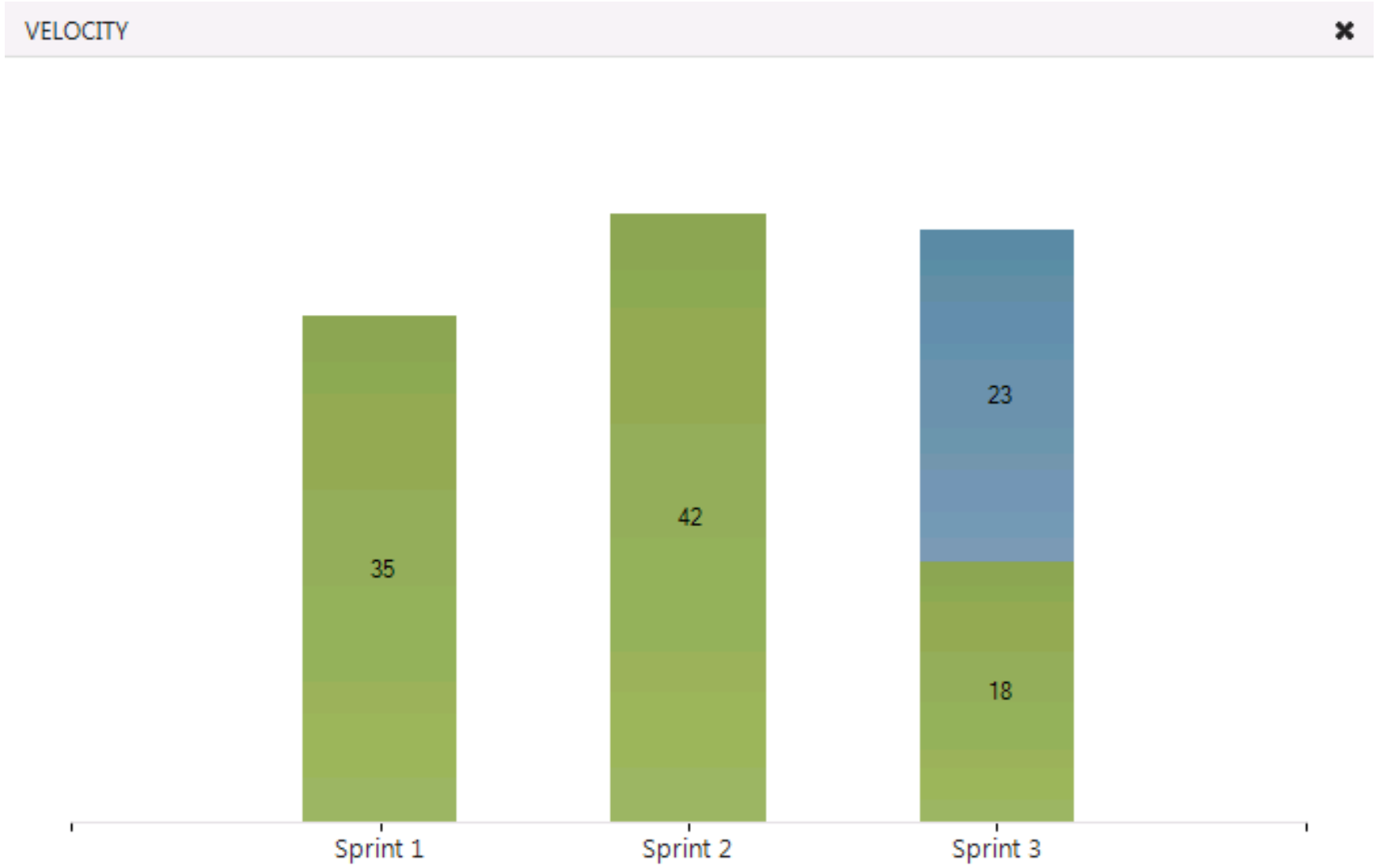
Board view by Task



Team member view



Velocity Chart



Forecasting

Forecasting based on velocity of **10**

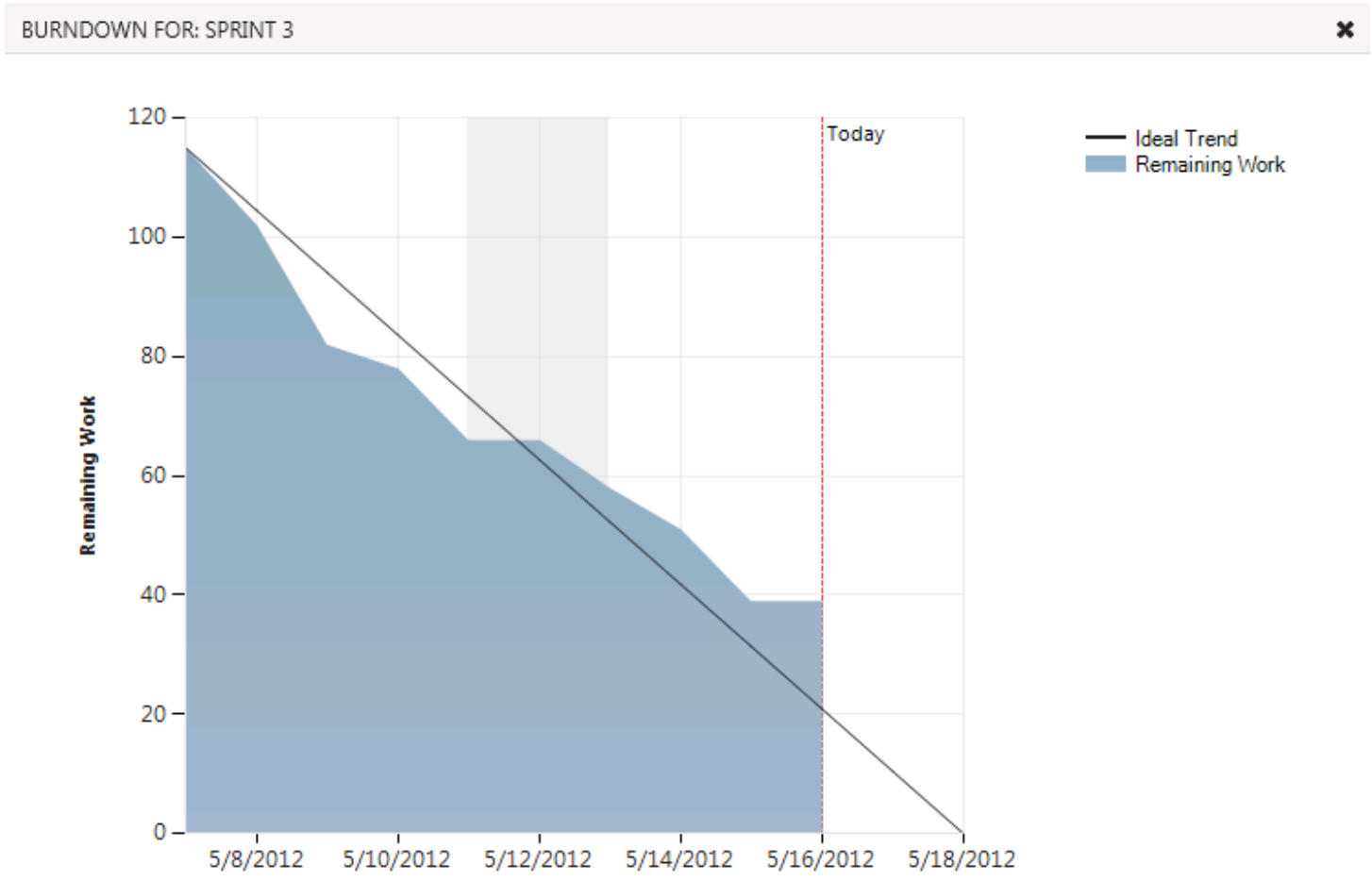
Forecast	Order	Title	State	Effort
Sprint 4	1	Customer should see weather-related outages on portal.	Approved	8
	2	Service rep can view service ticket details from the dashboard	New	8
Sprint 5	3	Technician can check on parts orders on Windows Phone.	New	3
Sprint 6	4	Technician can look for closest hardware store from Windows Phone.	New	3
	5	Technician can submit invoices on Windows Phone.	New	16
	6	Customer can view service invoices online.	New	4
	7	Customer can pay invoices online.	New	24
	8	Customer can opt-in/opt-out of paper billing.	New	8

Forecasting (Cont'd)

Forecasting based on velocity of 35

Forecast	Order	Title	State	Effort
	1	Customer should see weather-related outages on portal.	Approved	8
	2	Service rep can view service ticket details from the dashboard	New	8
	3	Technician can check on parts orders on Windows Phone.	New	3
Sprint 4	4	Technician can look for closest hardware store from Windows Phone.	New	3
	5	Technician can submit invoices on Windows Phone.	New	16
	6	Customer can view service invoices online.	New	4
Sprint 5	7	Customer can pay invoices online.	New	24
Sprint 6	8	Customer can opt-in/opt-out of paper billing.	New	8

Burndown Chart



The developer features

- Team Explorer
- Work Item Tracking
- Source Control Explorer
- Storyboarding, Code Review, Feedback
- Code Lens, Code Metrics

Nightly Fabrikam (Dev) | Source Control Explorer

- General
- Trigger
- Workspace
- Build Defaults
- Process
- Retention Policy

Select one of the following check-in triggers:

- (•) Manual - Check-ins do not trigger a new build
- ( ) Continuous Integration - Build each check-in
- ( ) Rolling builds - accumulate check-ins until the prior build finishes
  - [ ] Build no more often than every [ ] minutes.
- ( ) Gated Check-in - accept check-ins only if the submitted changes merge and build successfully
  - [ ] Merge and build up to [ ] submissions.
- ( ) Schedule - build every week on the following days
  - [x] Monday [x] Tuesday [x] Wednesday [x] Thursday
  - [x] Friday [ ] Saturday [ ] Sunday

Queue the build on the build controller at:

3:00 AM Pacific Standard Time (GMT -08:00)

- [ ] Build even if nothing has changed since the previous build

Misunderstood Requirements

Requirements don't reflect customer intent

Rework, long delivery cycles and frustration

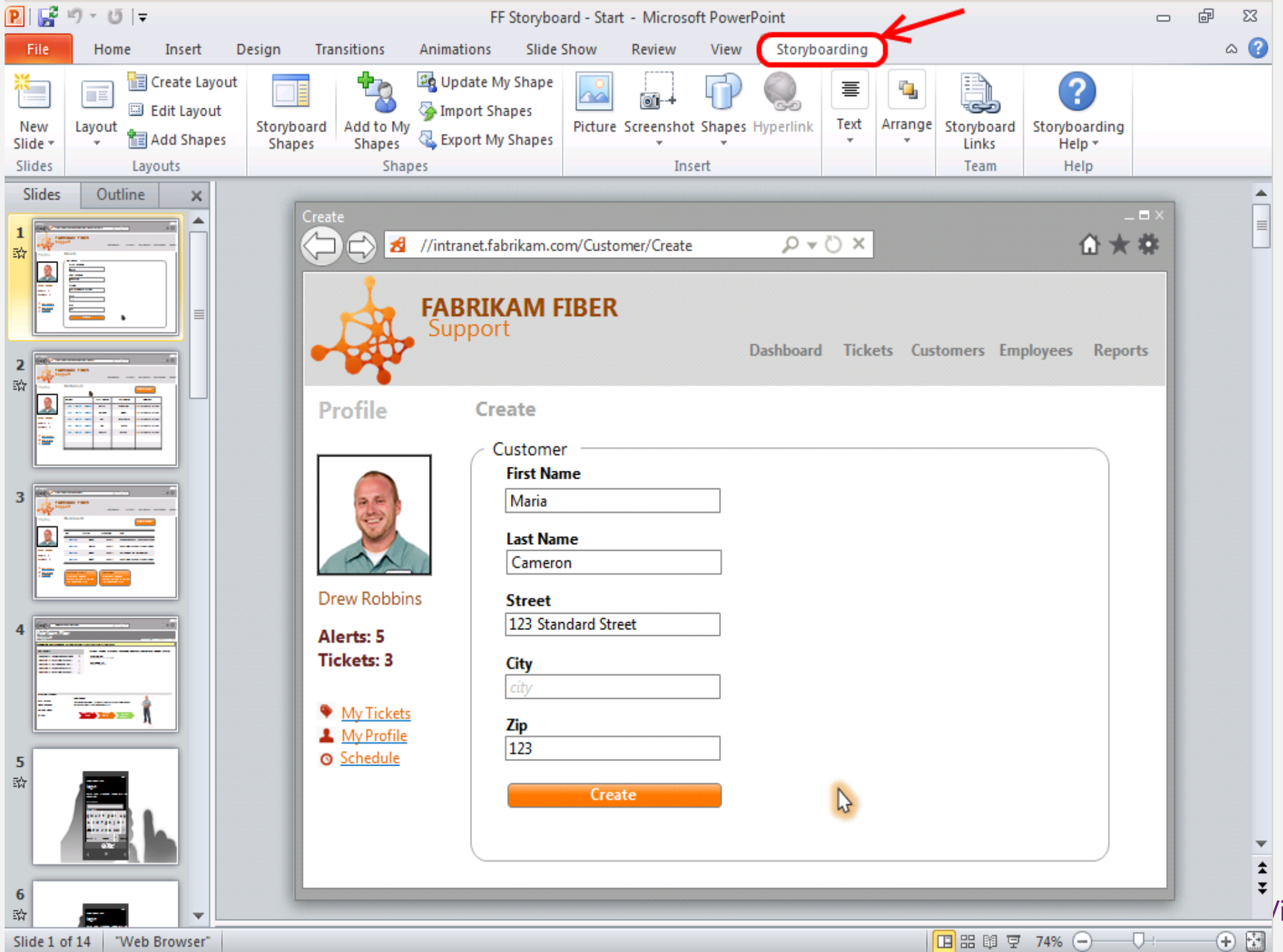
Storyboarding tool

Graphical mockups in new PowerPoint add-on reduce misunderstandings and clarify intent

Feedback Client

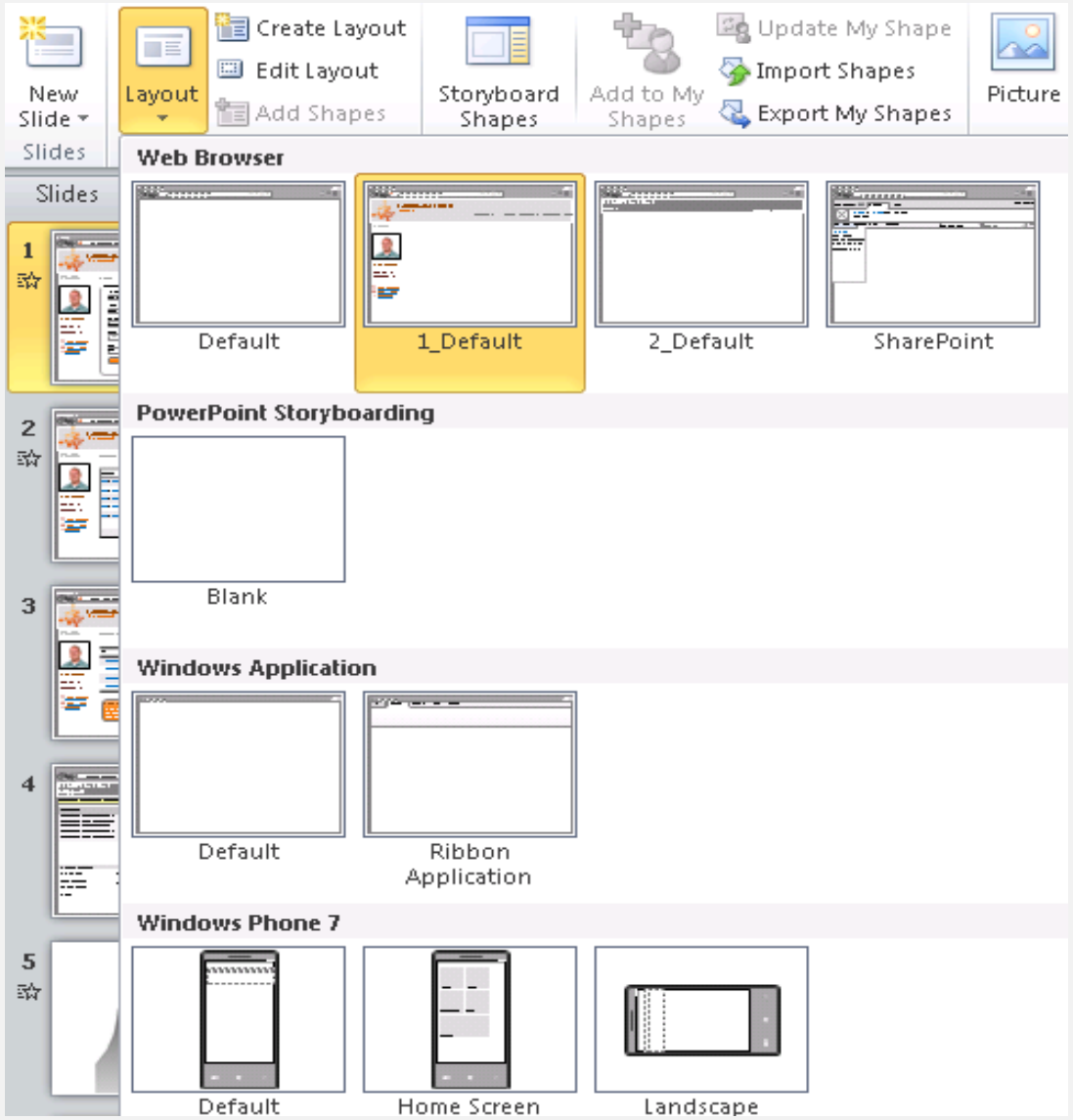
Stand-alone tool provides comprehensive feedback on any artifact, including story boards, directly to the development team

Requirement Gathering



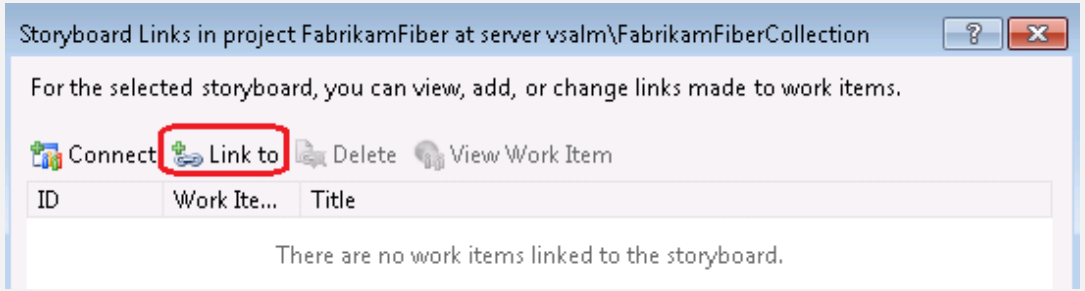
Visual Studio

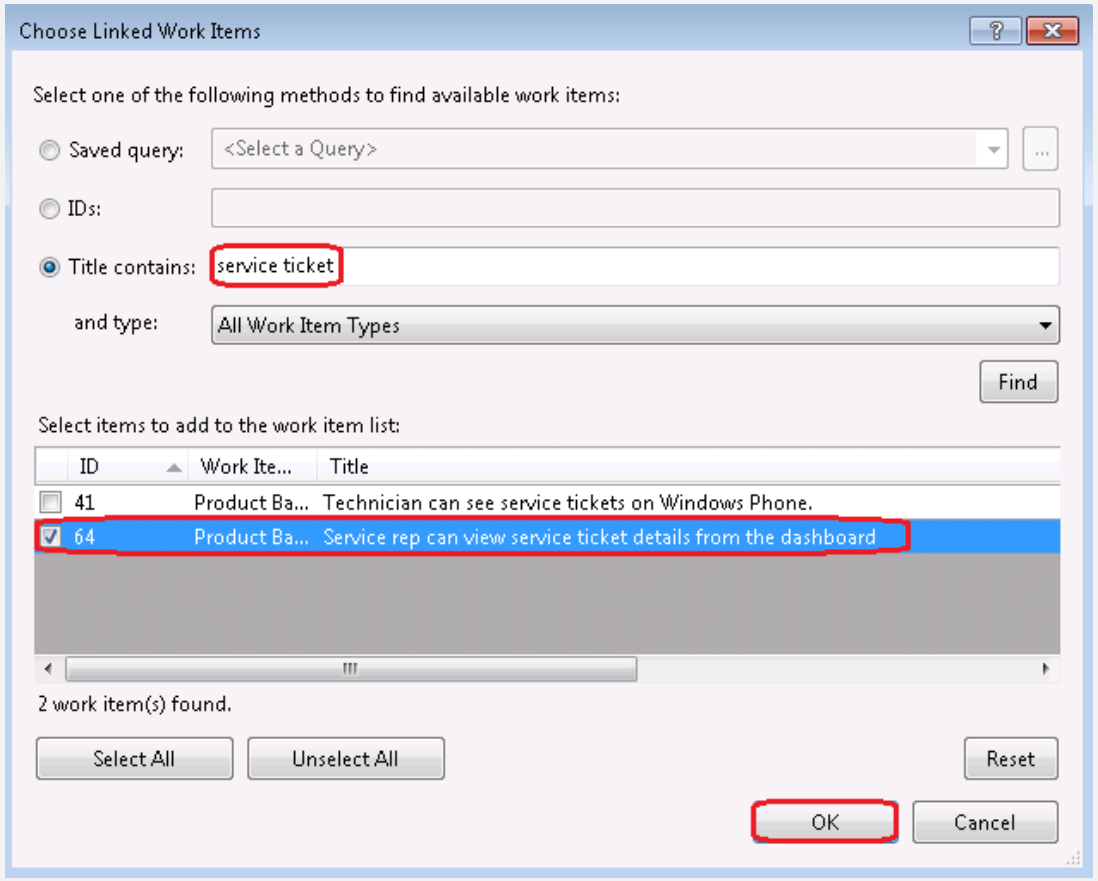
Create Own Shapes & Layout



Visual Studio

Link back to Work Item





Visual Studio

Unmet User Expectations

Delivered code fails to satisfy users

Failure to meet user needs, rework, long delivery cycles and customer dissatisfaction

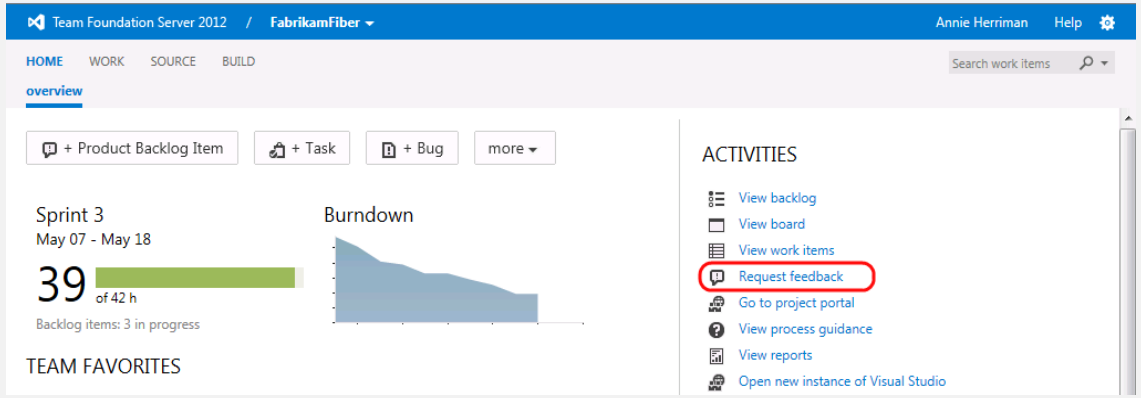
Exploratory testing

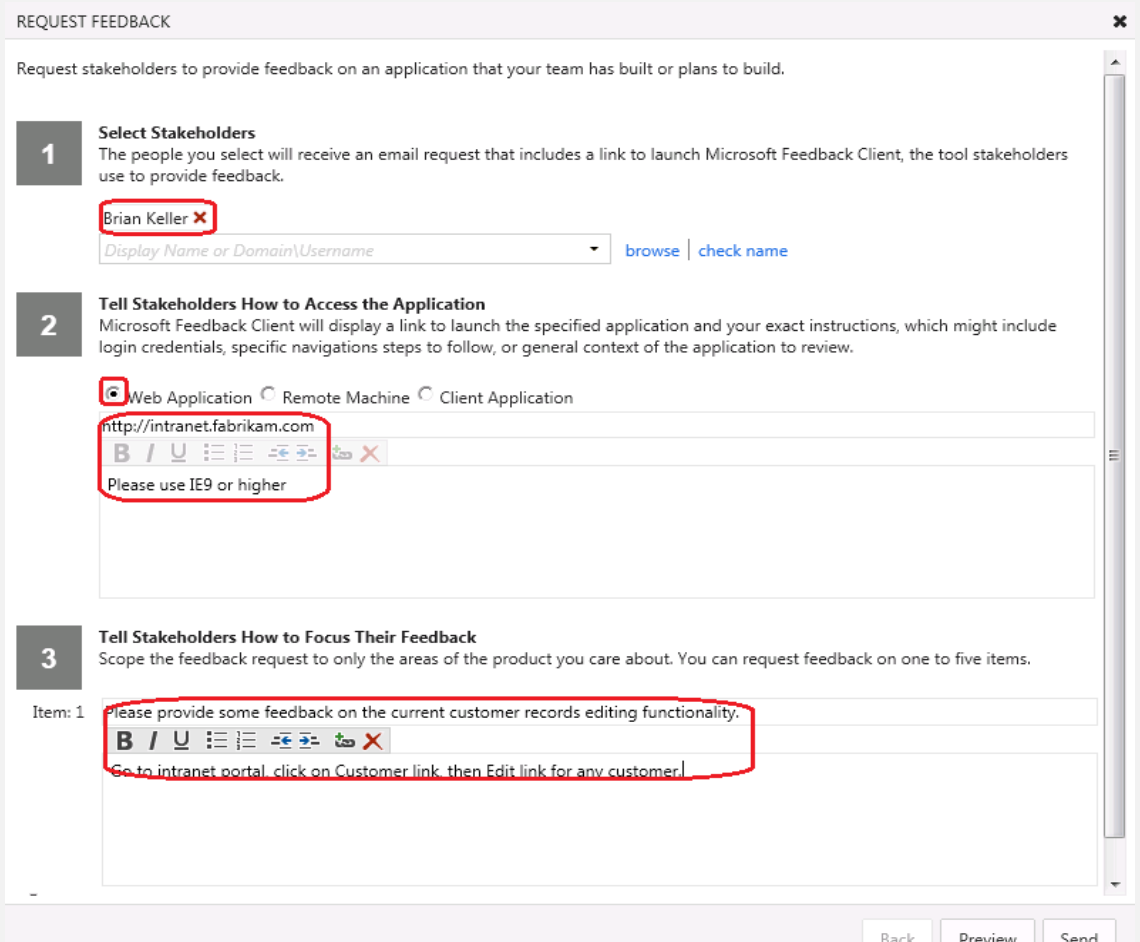
Ad-hoc exploratory tests help uncover usability and consistency issues, and result in actionable bugs and reusable test cases

Feedback Client

Stand-alone tool provides comprehensive feedback in-progress work, directly from users to the development team

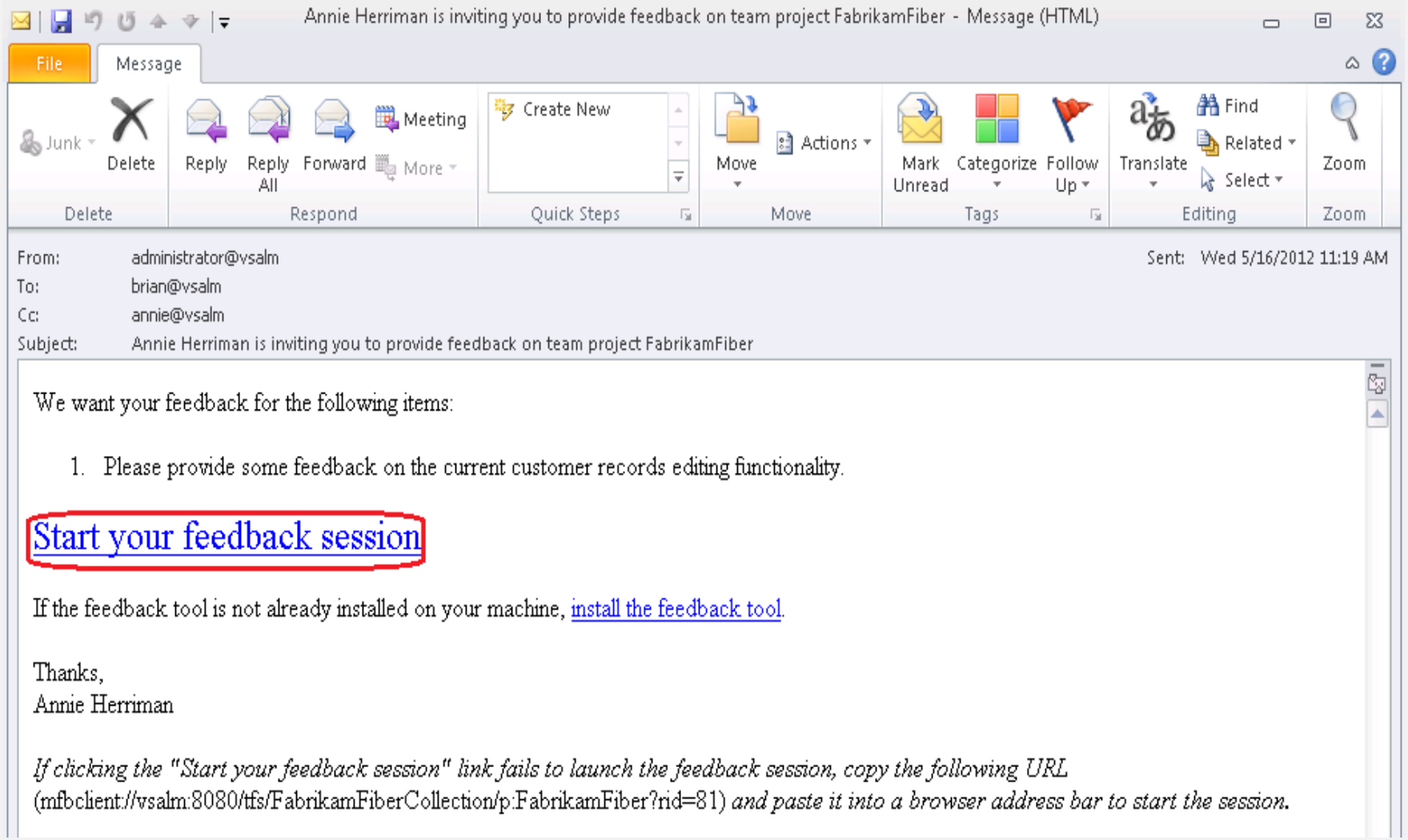
Request / Provide feedback





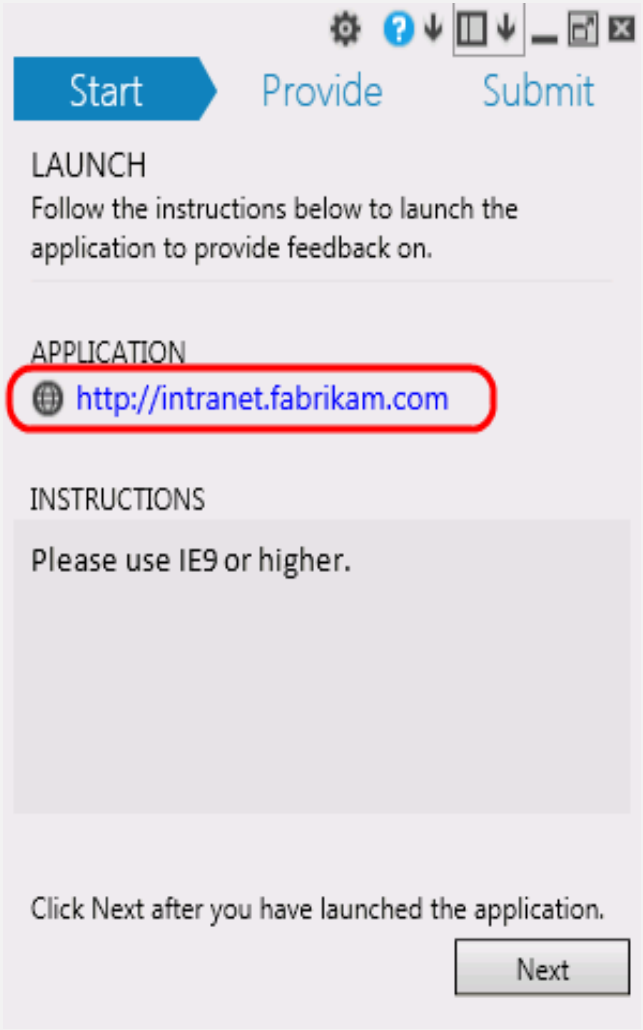
Visual Studio

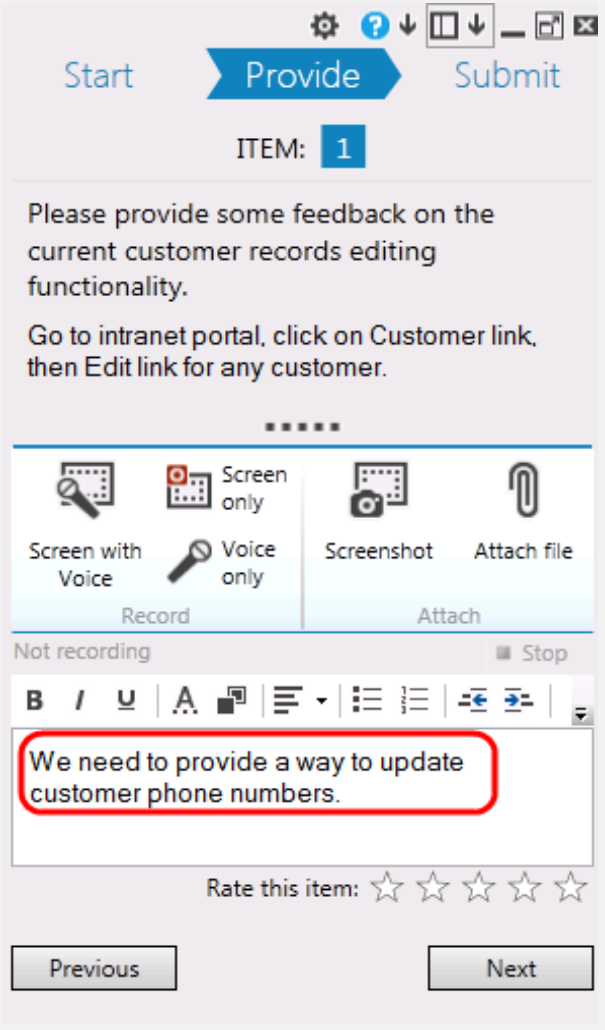
Request sent by email



Visual Studio

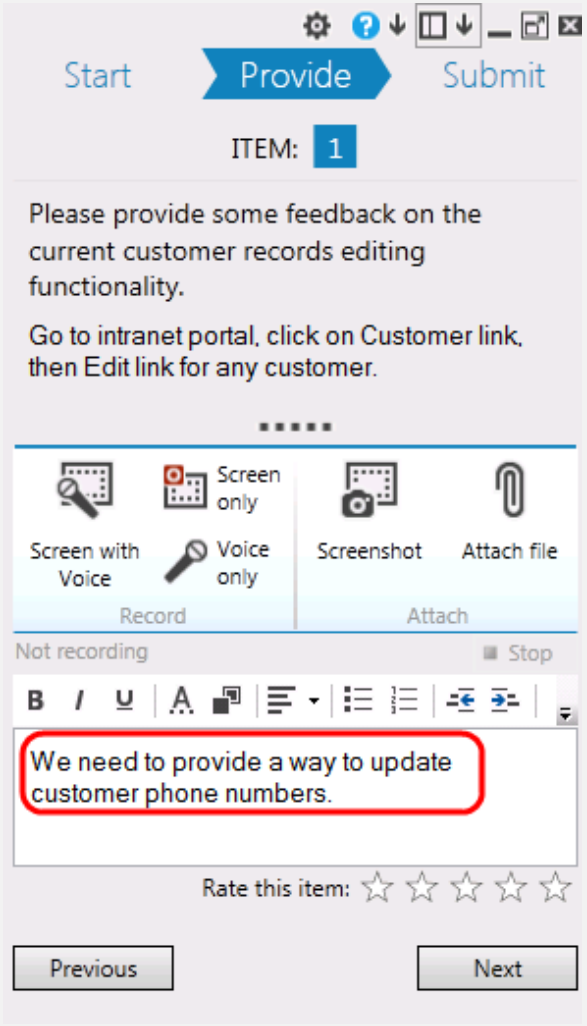
Feedback Client

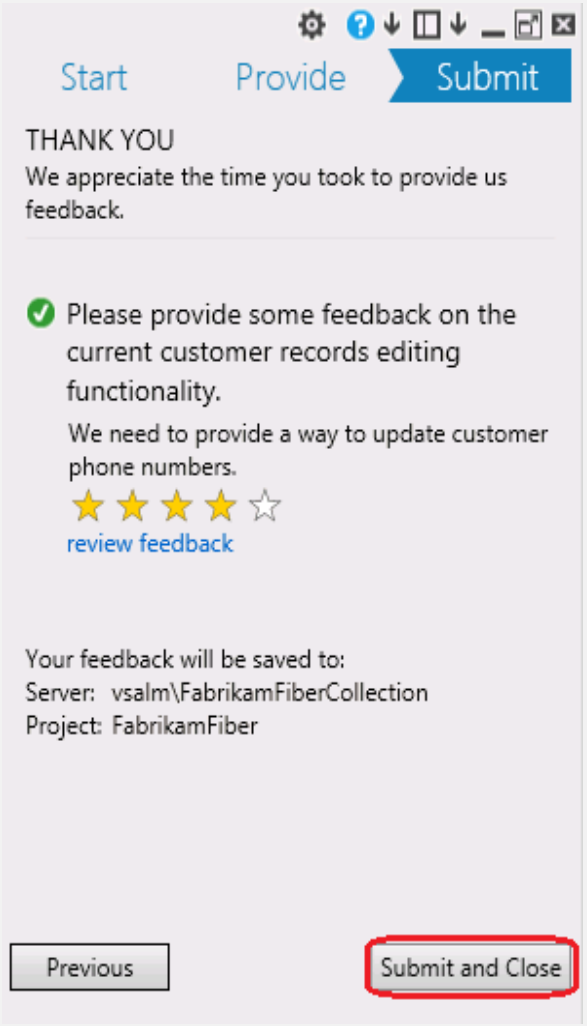




Visual Studio

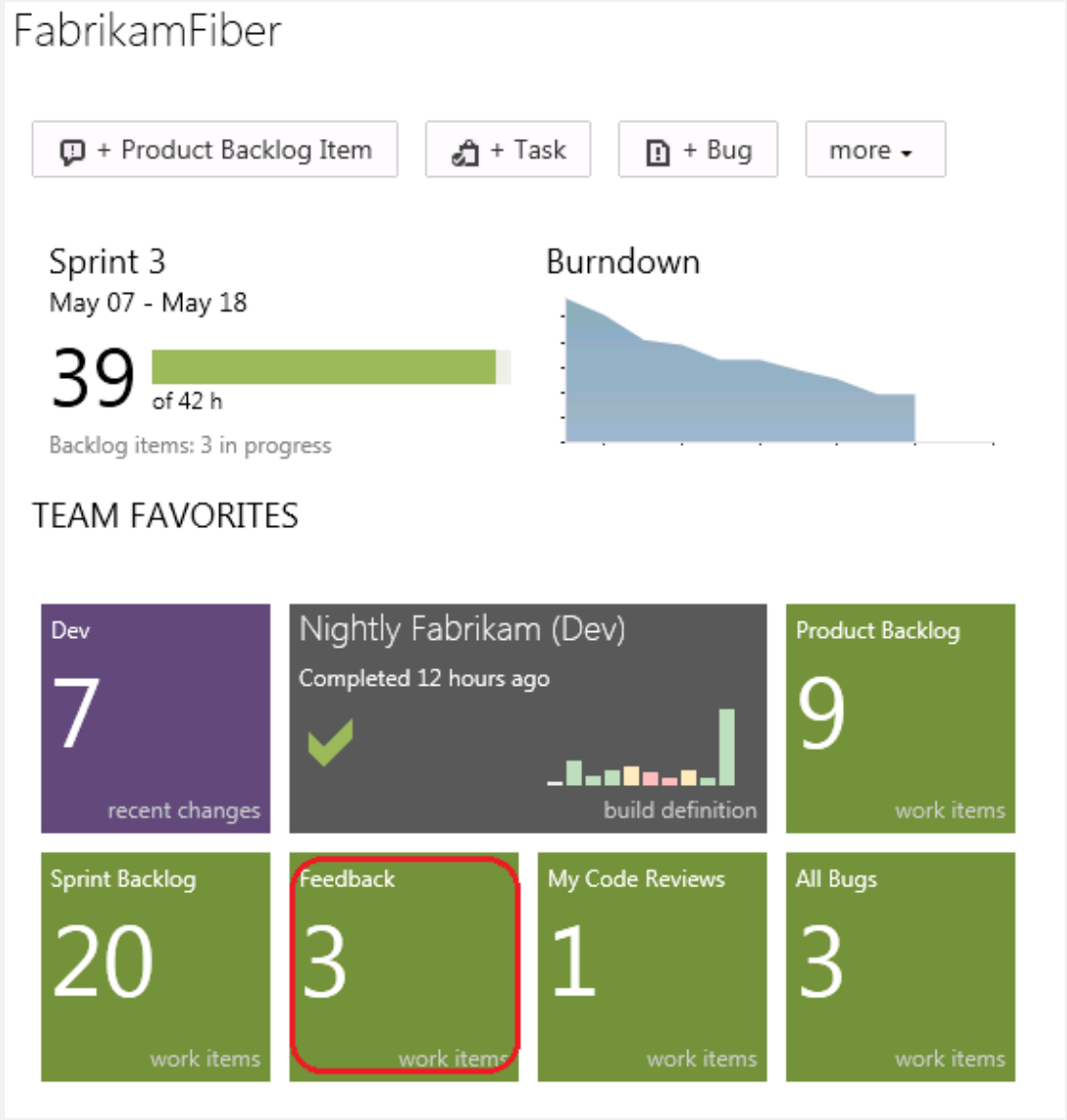
Providing Feedback





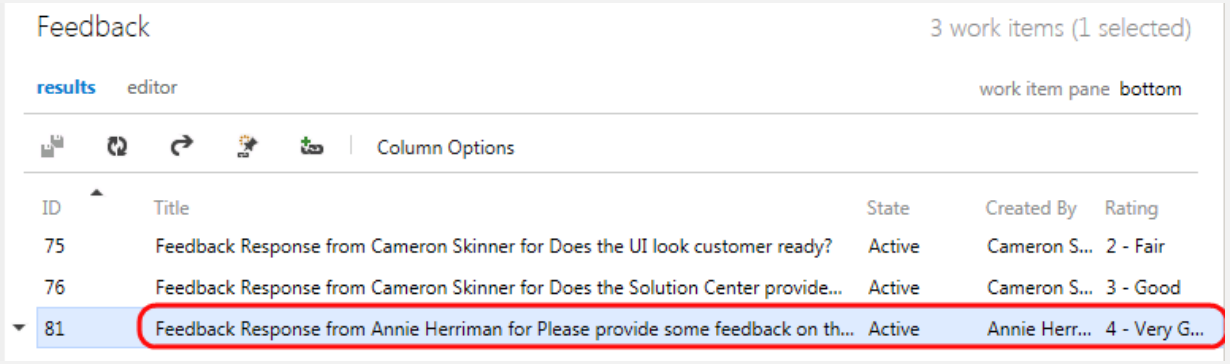
Visual Studio

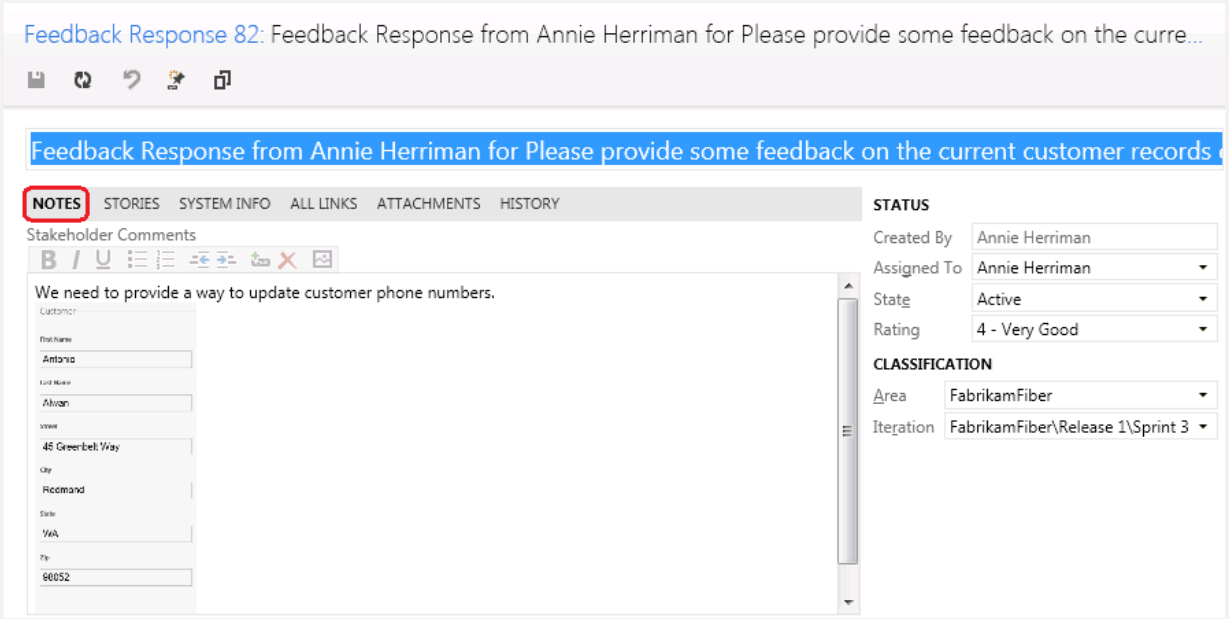
New Feedback



Visual Studio

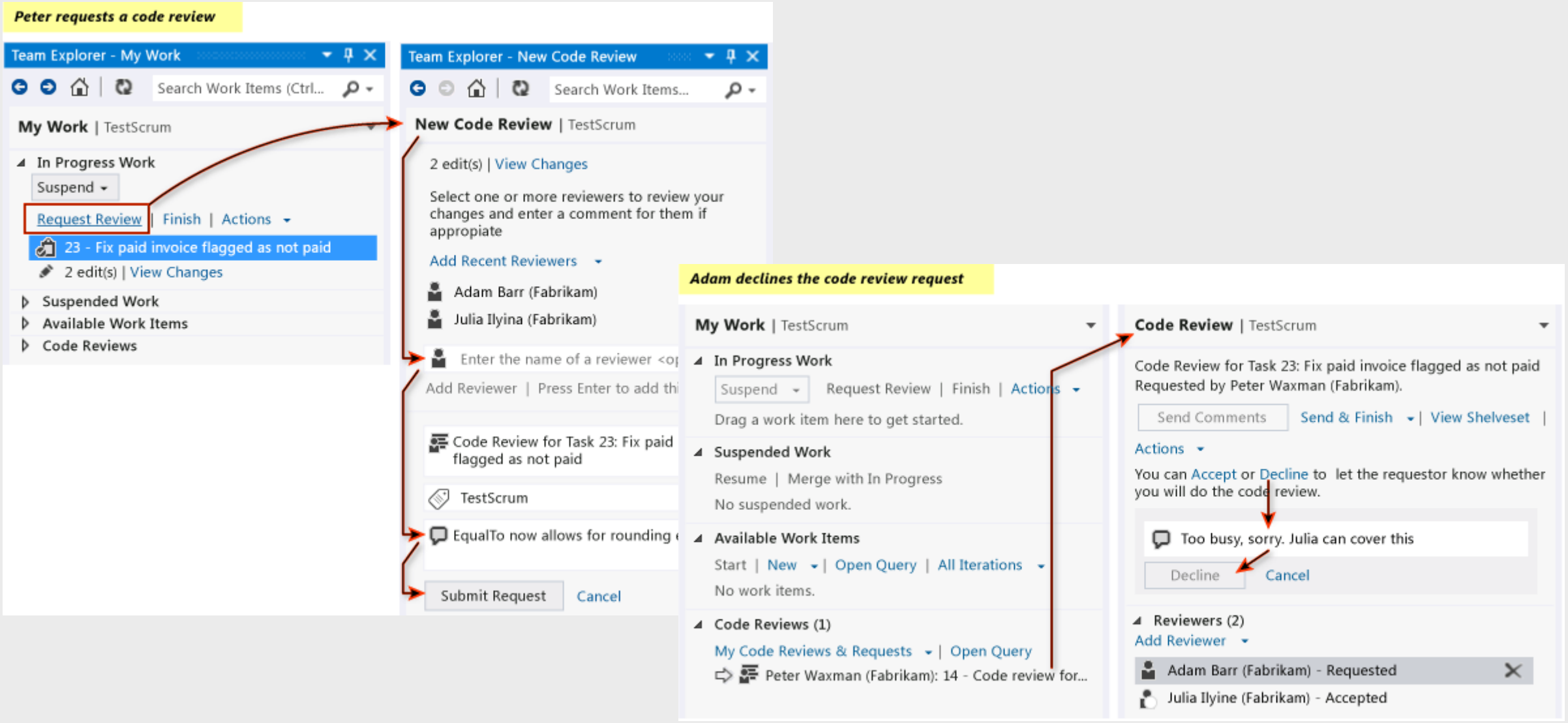
Reviewing User Feedback





Visual Studio

Code Review

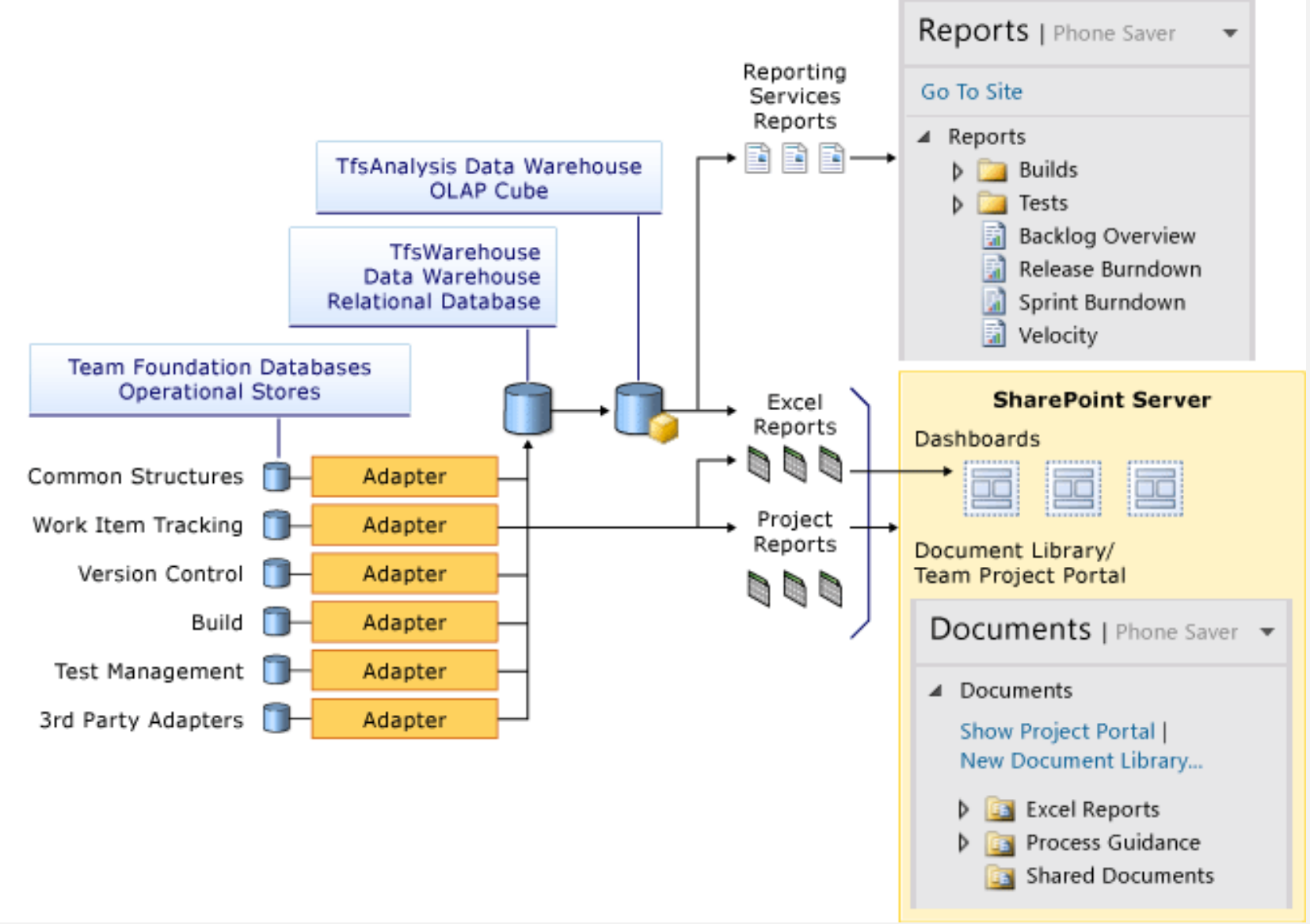

Peter requests a code review
Team Explorer - My Work
Search Work Items (Ctrl...
My Work | TestScrum
In Progress Work
Suspend
Request Review | Finish | Actions
23 - Fix paid invoice flagged as not paid
2 edit(s) | View Changes
Suspended Work
Available Work Items
Code Reviews
Team Explorer - New Code Review
Search Work Items...
New Code Review | TestScrum
2 edit(s) | View Changes
Select one or more reviewers to review your changes and enter a comment for them if appropiate
Add Recent Reviewers
Adam Barr (Fabrikam)
Julia Ilyina (Fabrikam)
Enter the name of a reviewer
Add Reviewer | Press Enter to add
Code Review for Task 23: Fix paid flagged as not paid
TestScrum
EqualTo now allows for rounding
Submit Request
Cancel
Adam declines the code review request
My Work | TestScrum
In Progress Work
Suspend
Request Review | Finish | Actions
Drag a work item here to get started.
Suspended Work
Resume | Merge with In Progress
No suspended work.
Available Work Items
Start | New | Open Query | All Iterations
No work items.
Code Reviews (1)
My Code Reviews & Requests | Open Query
Peter Waxman (Fabrikam): 14 - Code review for...
Code Review | TestScrum
Code Review for Task 23: Fix paid invoice flagged as not paid
Requested by Peter Waxman (Fabrikam).
Send Comments
Send & Finish | View Shelveset
Actions
You can Accept or Decline to let the requestor know whether you will do the code review.
Too busy, sorry. Julia can cover this
Decline
Cancel
Reviewers (2)
Add Reviewer
Adam Barr (Fabrikam) - Requested
Julia Ilyine (Fabrikam) - Accepted

Manager Perspective

- Team Web Access
- Team Project Portal
- Excel & Project Integration
- Reporting
- Team Room

TFS Reporting

TFS Reporting



Visual Studio

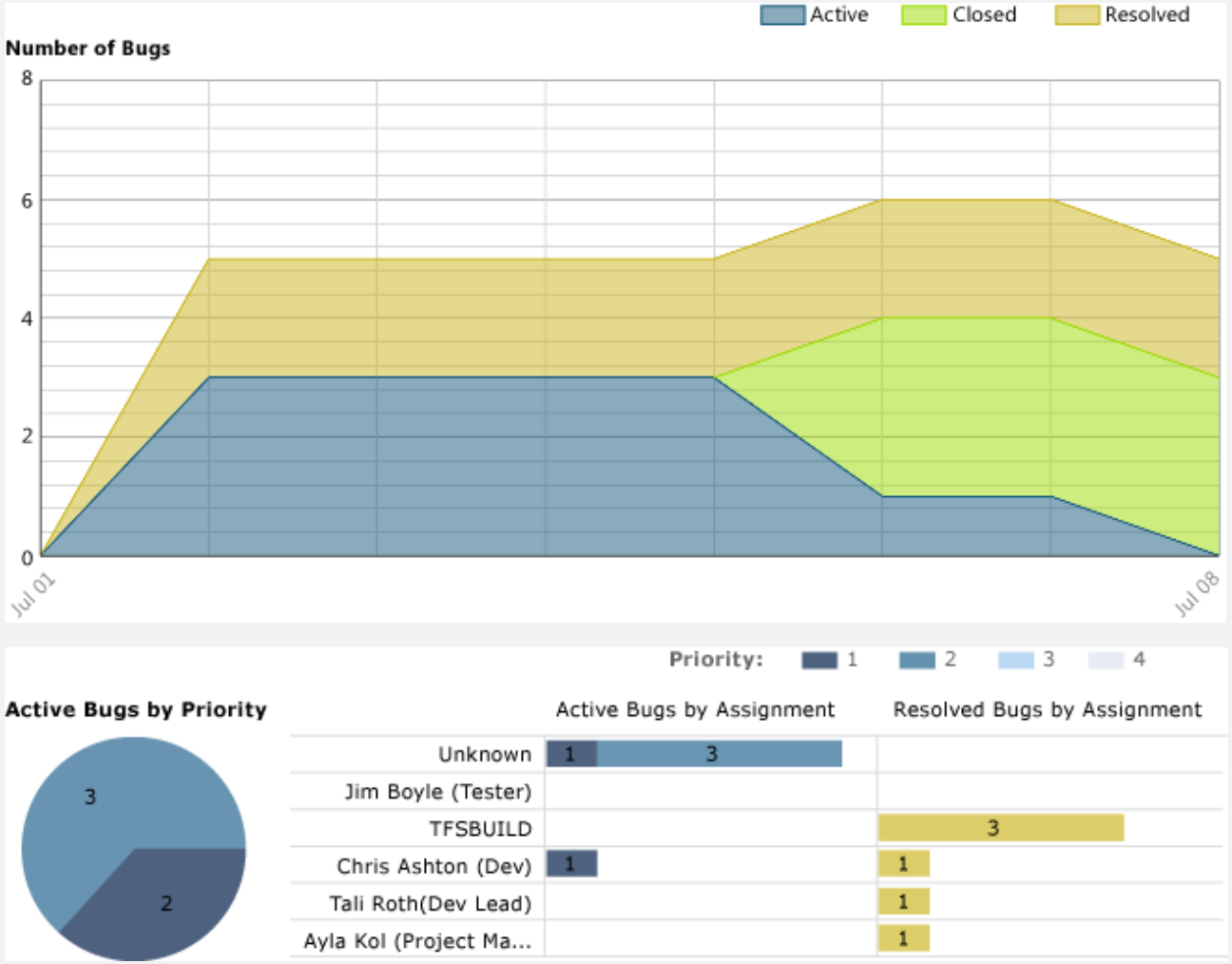
Tasks	Report names and related topics
Monitor bug activity, reactivations, and trends. You can use bug reports to track the bugs that the team is finding and the progress that the team is making toward fixing them.	• Bug Status Report • Bug Trends Report • Reactivations Report
Monitor build activity, success, and trends. You can use build reports to track the quality and success of your team's builds over time.	• Build Quality Indicators Report • Build Success Over Time Report • Build Summary Report
Track project health, team burn rate, and story and task completion. You can use the Requirements Progress report to review the level of effort that the team has spent on each user story that the team is implementing. By using this report, you can quickly determine whether any work was recently completed on each story and what work is remaining. You can use the Requirements Overview report to help you track how far each user story has been implemented and tested. You can review this report daily or weekly to monitor the progress of the team during an iteration.	• Burndown and Burn Rate Report (CMMI) • Remaining Work Report • Status on All Iterations Report • Requirements Progress Report (CMMI) • Requirements Overview Report (CMMI)
Determine added work. You can use the Unplanned Work report to determine how much work the team added to an iteration after it started.	• Unplanned Work
Monitor testing activity. You can use the test reports to track the team's progress toward developing Test Cases and to determine how well they cover the Requirements.	• Test Case Readiness Report • Test Plan Progress Report

Visual Studio

Monitor bug activity, reactivations, and trends

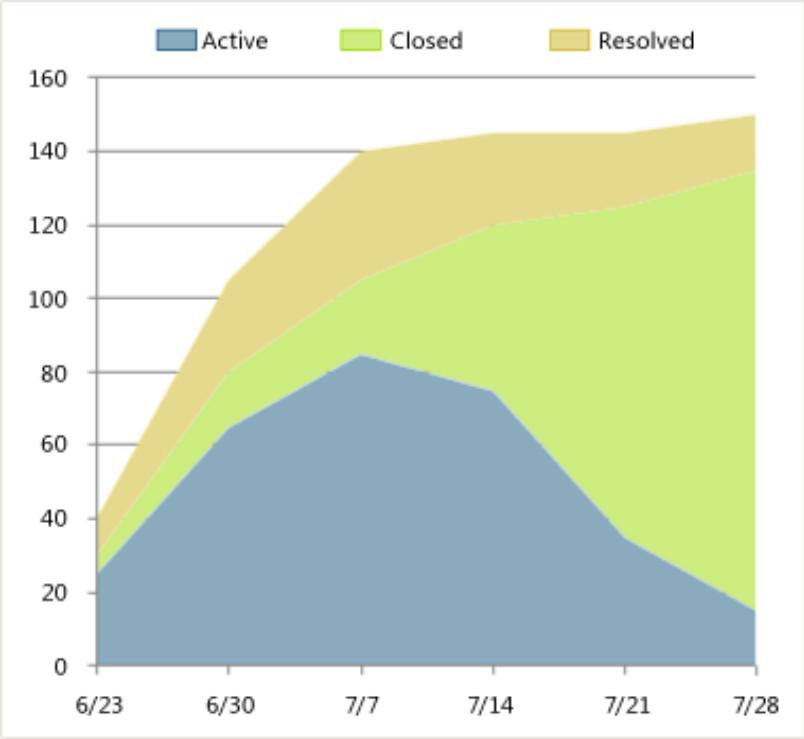
- Is the team fixing bugs quickly enough to finish on time?
- Is the team fixing high priority bugs first?
- What is the distribution of bugs by priority and severity?
- How many bugs are assigned to each team member?

Bug Status Report

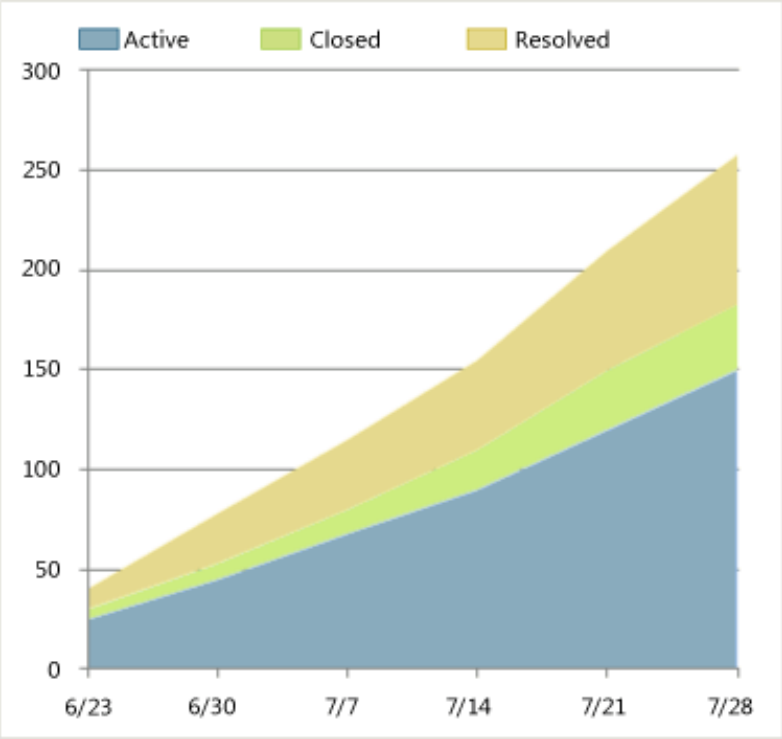


Visual Studio

Information	Description
Number of Bugs	A visual representation of the cumulative count of all bugs, grouped by their state.
Active Bugs by Priority/Severity	A pie chart that depicts the number of bugs that are still active, grouped by priority or severity.
Active Bugs by Assignment	A horizontal bar chart with the total count of bugs that each team member has assigned to them in the active state, grouped by priority or severity.
Resolved Bugs by Assignment	A horizontal bar chart with the total count of bugs that each team member has assigned to them in the resolved state, grouped by priority or severity.



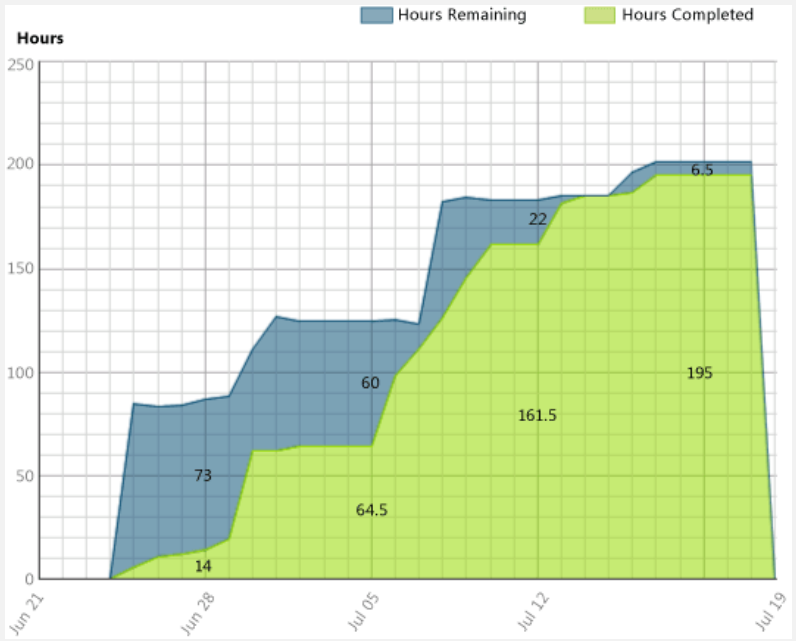
Healthy

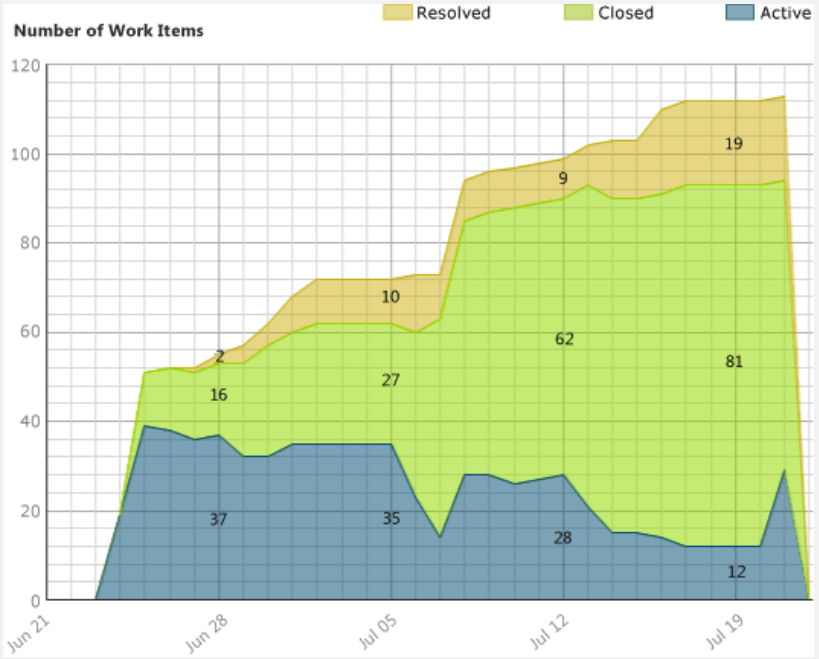


Unhealthy

Track project health, task

- What is the cumulative flow of work?
- Is the team likely to finish the iteration on time?
- Is the amount of work or number of work items in the iteration growing?
- Does the team have too much work in progress?
- How is the team doing in estimating work for the iteration?



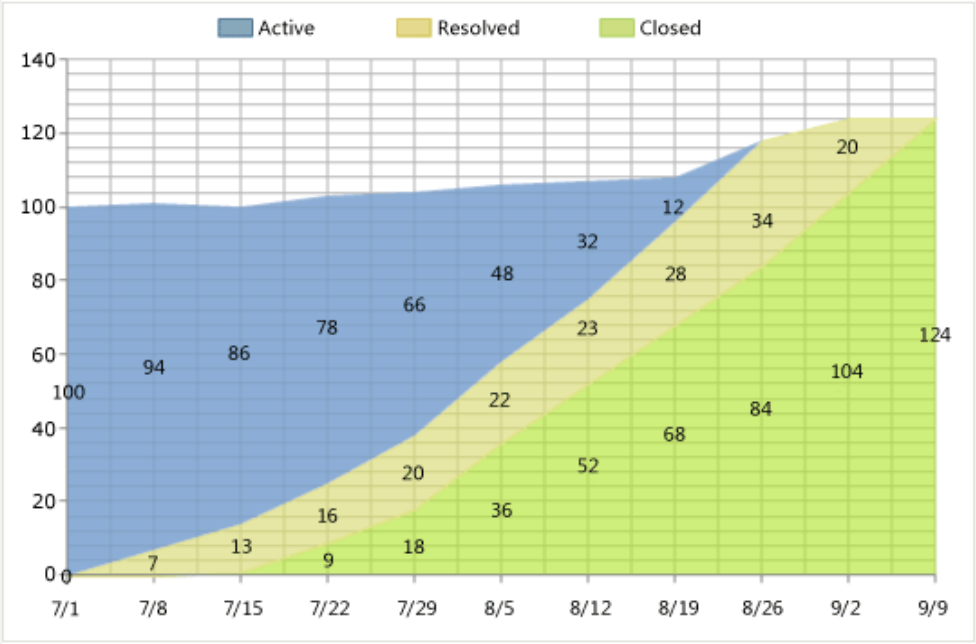


Hours of work

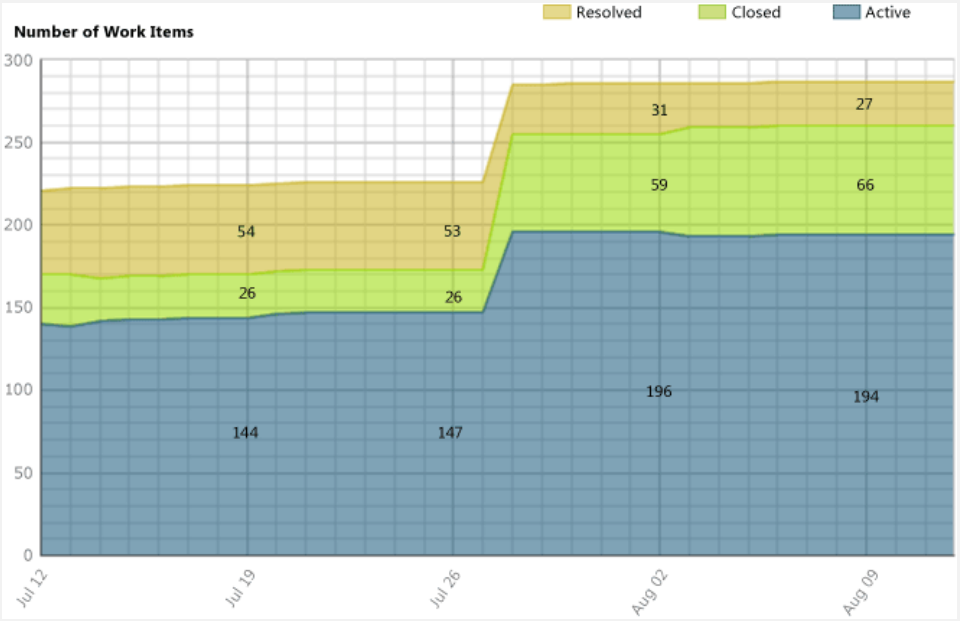
Of Work Items

Visual Studio

Hours	Number of Work Items
• **Hours Remaining**: The cumulative value of all remaining hours for all tasks. • **Hours Completed**: The cumulative value of all completed hours for all tasks.	• **Active**: The cumulative value of all stories, tasks, and bugs that are in the **Active** (blue) state. • **Resolved**: The cumulative value of all stories or bugs and that are in the **Resolved** (gold) state. • **Closed**: The cumulative value of all stories, tasks, and bugs in the **Closed** (green) state.



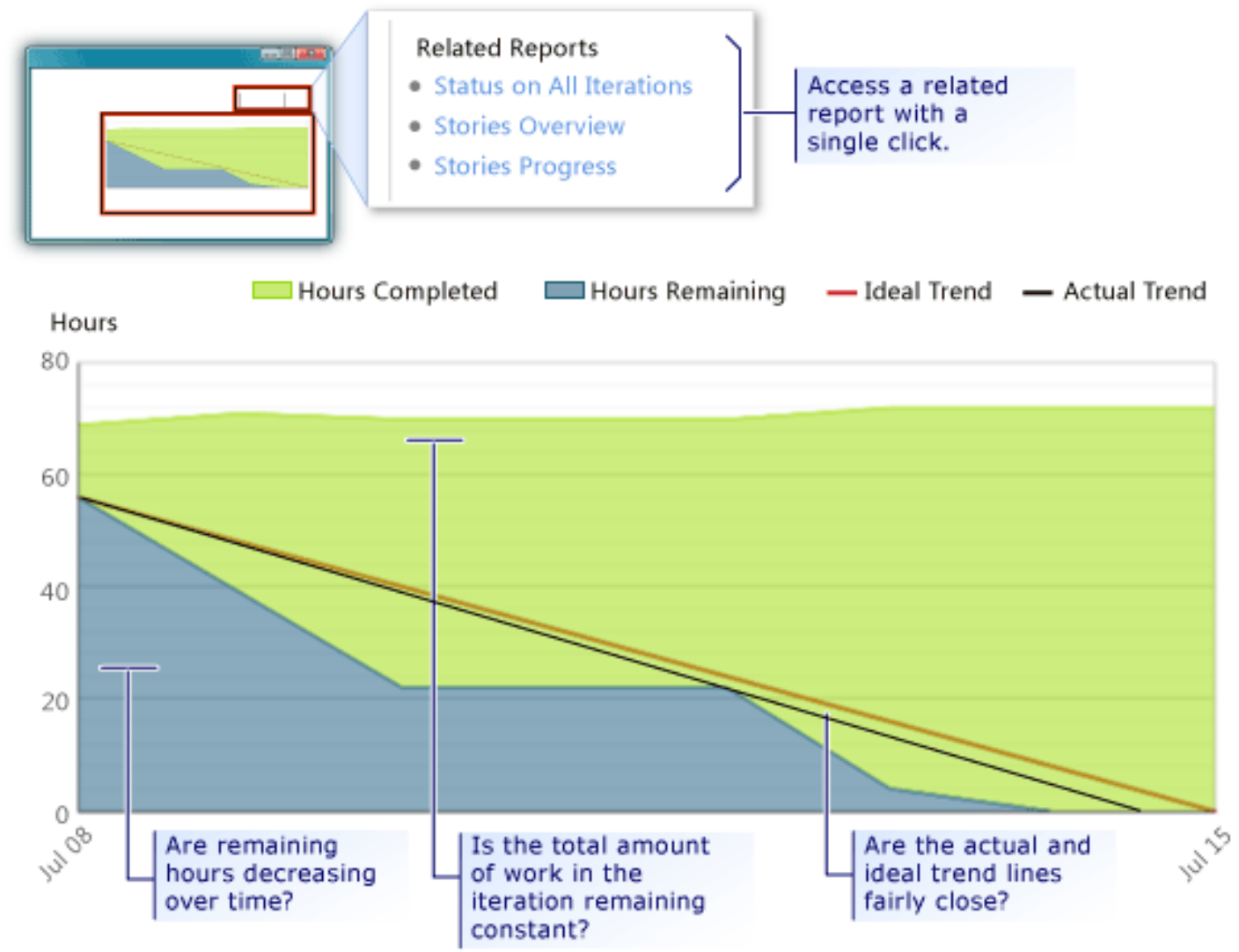
Healthy



Unhealthy

Visual Studio

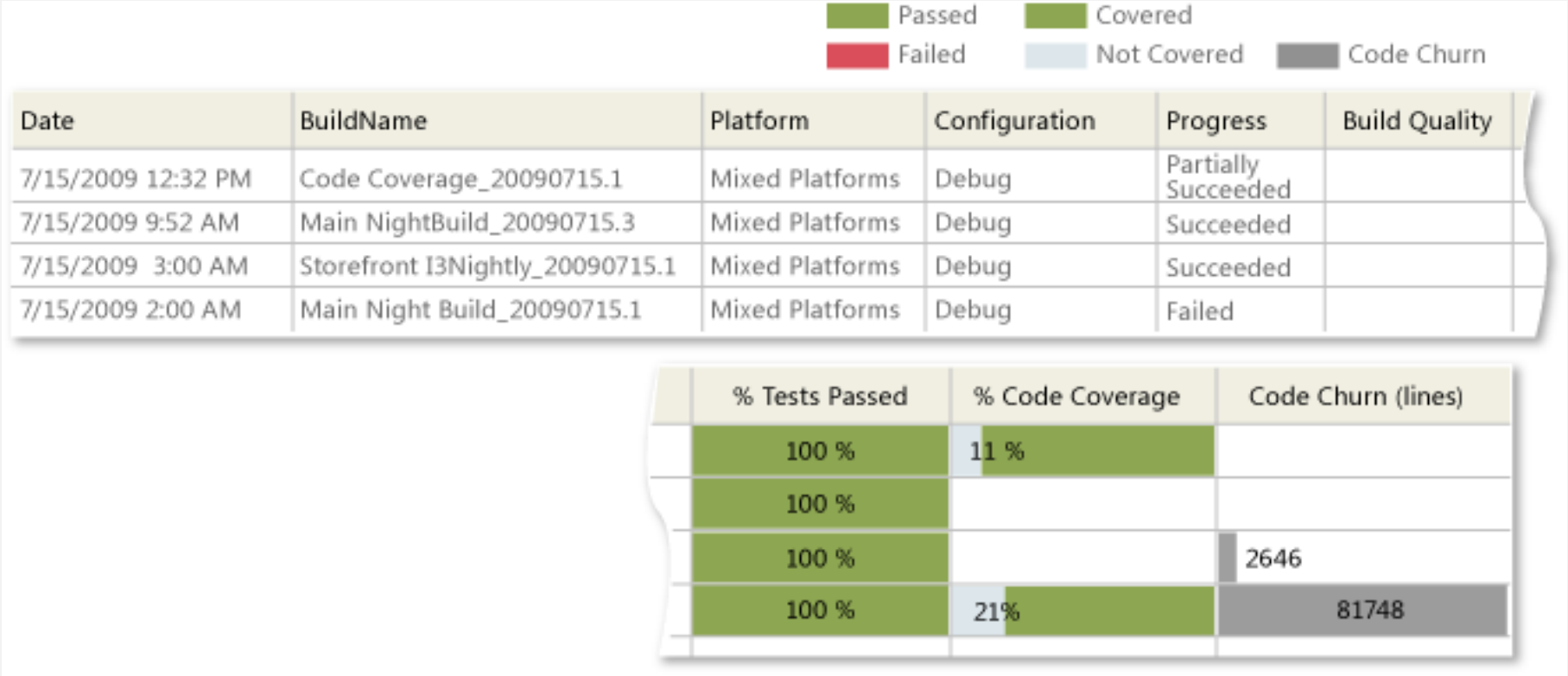
Burn down rate



Visual Studio

Build reports

- What is the status of all builds over time?
- Which builds succeeded?
- Which builds have a significant number of changes to the code?
- How much of the code was executed by the tests?
- Which builds are ready to install?



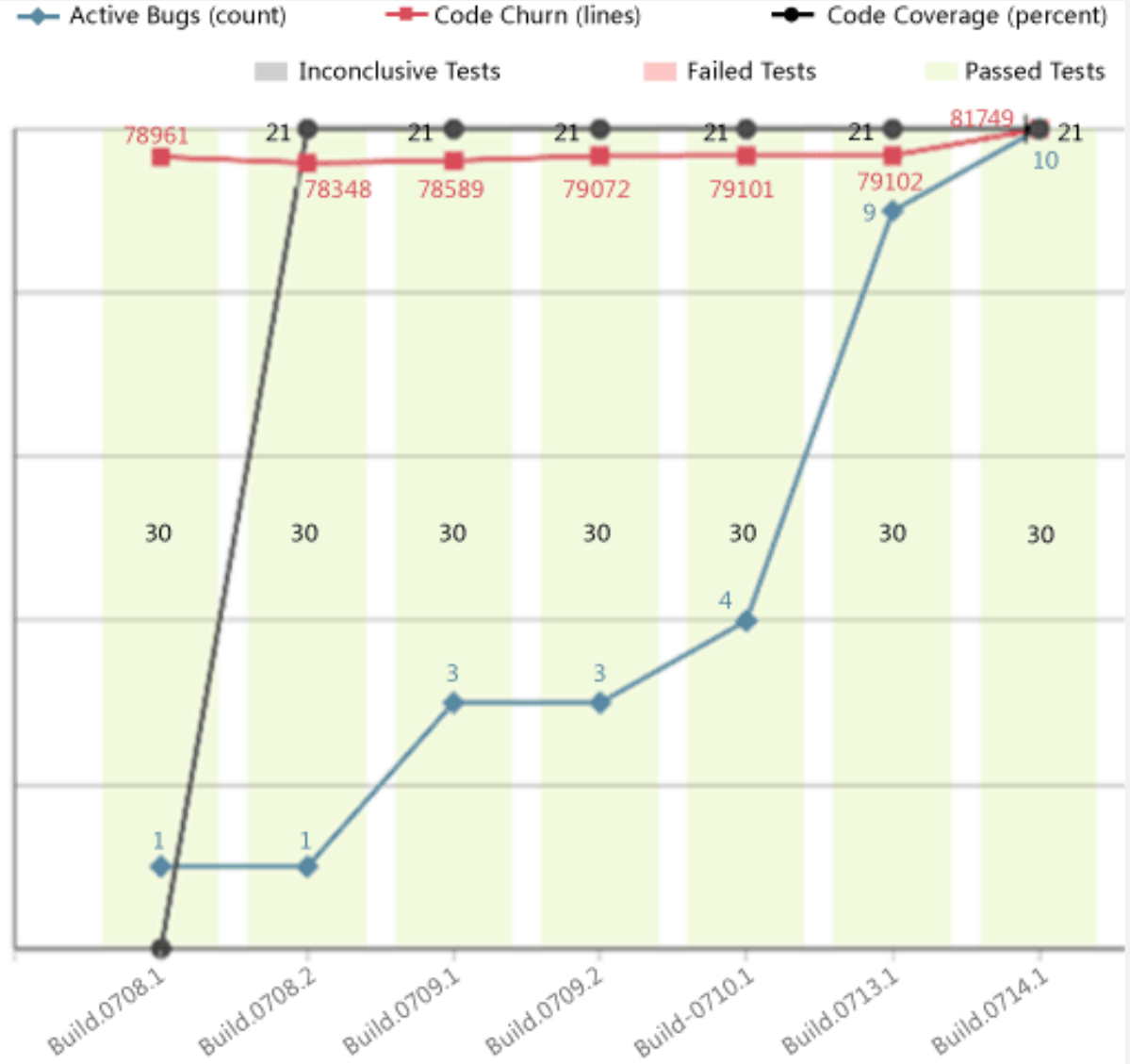
Date	BuildName	Platform	Configuration	Progress	Build Quality
7/15/2009 12:32 PM	Code Coverage_20090715.1	Mixed Platforms	Debug	Partially Succeeded	
7/15/2009 9:52 AM	Main NightBuild_20090715.3	Mixed Platforms	Debug	Succeeded	
7/15/2009 3:00 AM	Storefront I3Nightly_20090715.1	Mixed Platforms	Debug	Succeeded	
7/15/2009 2:00 AM	Main Night Build_20090715.1	Mixed Platforms	Debug	Failed	

% Tests Passed	% Code Coverage	Code Churn (lines)
100 %	11 %	
100 %		
100 %		2646
100 %	21%	81748

Visual Studio

Quality indicator	Description
Build Progress	Specifies the status of the build. A build can be in one of the following states: • **Failed**. The build failed to compile or tests failed to pass. • **Partially Succeeded**. Only some portions of the build successfully compiled. • **Stopped**. The build was manually stopped. • **Succeeded**. The build successfully compiled, and tests ran.
Build Quality	Specifies a manually assigned assessment of the quality of the build. You can add or remove the build qualities that are defined for your team project. For more information, see Add or Remove Build Quality Values. The column is empty if the build quality has not been rated.
% Tests Passed	Displays a horizontal stacked bar chart that lists the percentage of tests that passed superimposed on a green bar. The remaining bar segment is red, which indicates the percentage of tests that failed. The total length of the chart always equals the width of the column.
% Code Coverage	Displays a horizontal stacked bar chart that lists the percentage of code that was covered superimposed on a green bar. The remaining bar segment is light blue, which indicates the percentage of code that was not tested in the build. The total length of the chart always equals the width of the column.
% Code Churn (lines)	Displays a horizontal bar chart that lists the percentage of code churn superimposed on a gray bar. The code churn is calculated by determining the number of lines of code that the team has added, deleted, or modified divided by the total number of lines in the build. The bar length is proportionate to the percentage figure, scaled across the report so that the maximum amount of code churn across all builds equals the width of the column.

Build Quality Report



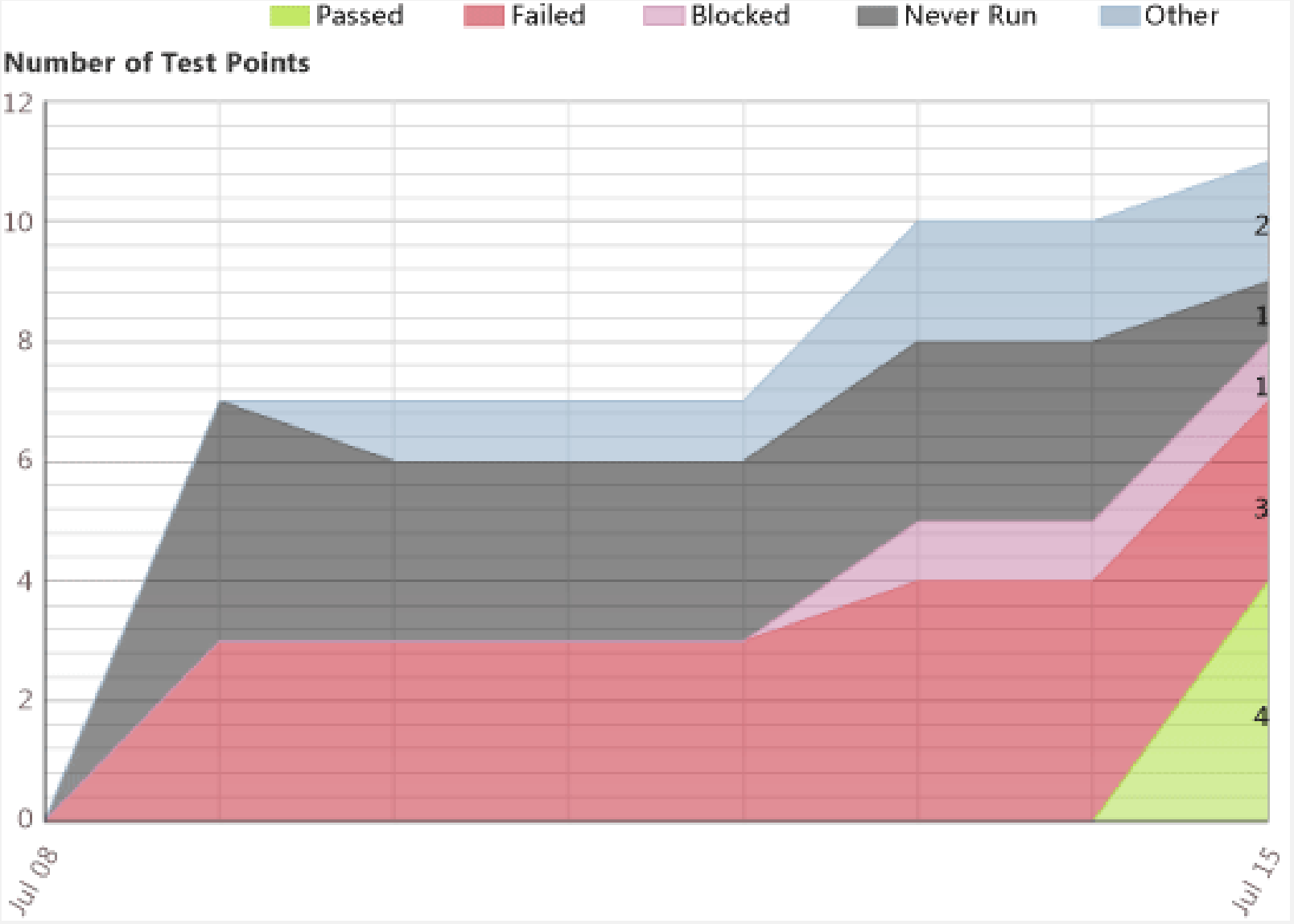
Visual Studio

Quality indicator	Description
Active Bugs (count)	Line chart that depicts the number of bugs that were active at the time of the build. **Note** Bugs are not explicitly associated with builds. Some of the bugs counted might not affect the builds that appear in the chart. You can use the Area parameter to filter bugs by product area. This technique might show bugs that are most likely to affect the builds in the report.
Code Churn (lines)	Line chart that depicts the number of lines of code that the team added, removed, and changed in the checkins before the build. The code churn is calculated by determining the number of lines of code that have been added, deleted, or modified to the build divided by the total lines in the build.
Code Coverage (percent)	Line chart that depicts the percentage of code that the tests cover.
Inconclusive Tests	Gray portion of the stacked bar chart, which indicates the number of tests that did not succeed or were paused. If the build did not succeed, the tests are either not counted or counted as inconclusive.
Failed Tests	Red portion of the stacked bar chart, which indicates the number of tests that failed for the build.
Passed Tests	Green portion of the stacked bar chart, which indicates the number of tests that passed for the build.

Test Reports

- How much testing has the team completed?
- Is the team likely to finish the testing on time?
- How many tests are left to be run?
- How many tests are passing?
- How many tests are failing?
- How many tests are blocked?

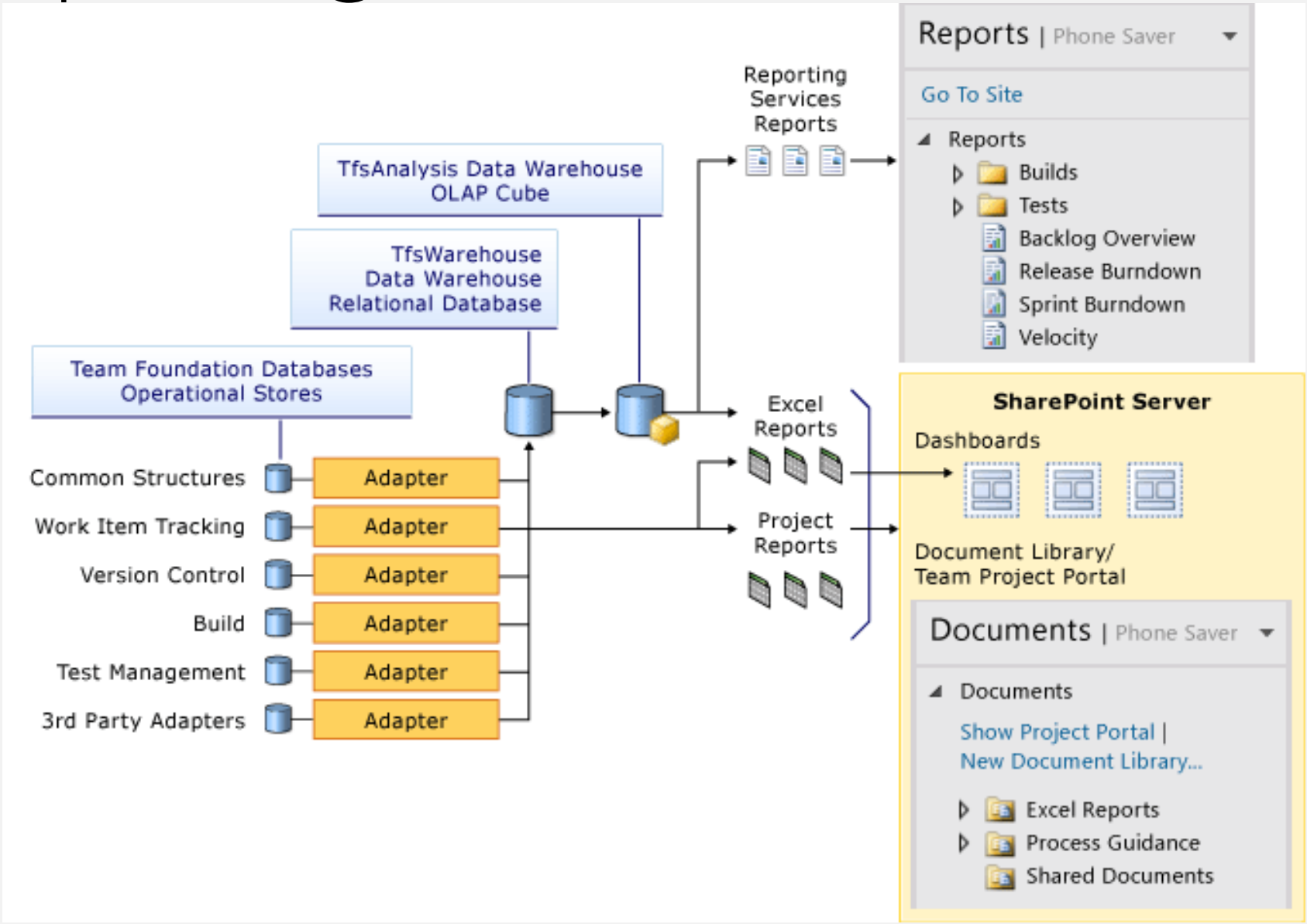
Test Plan Progress



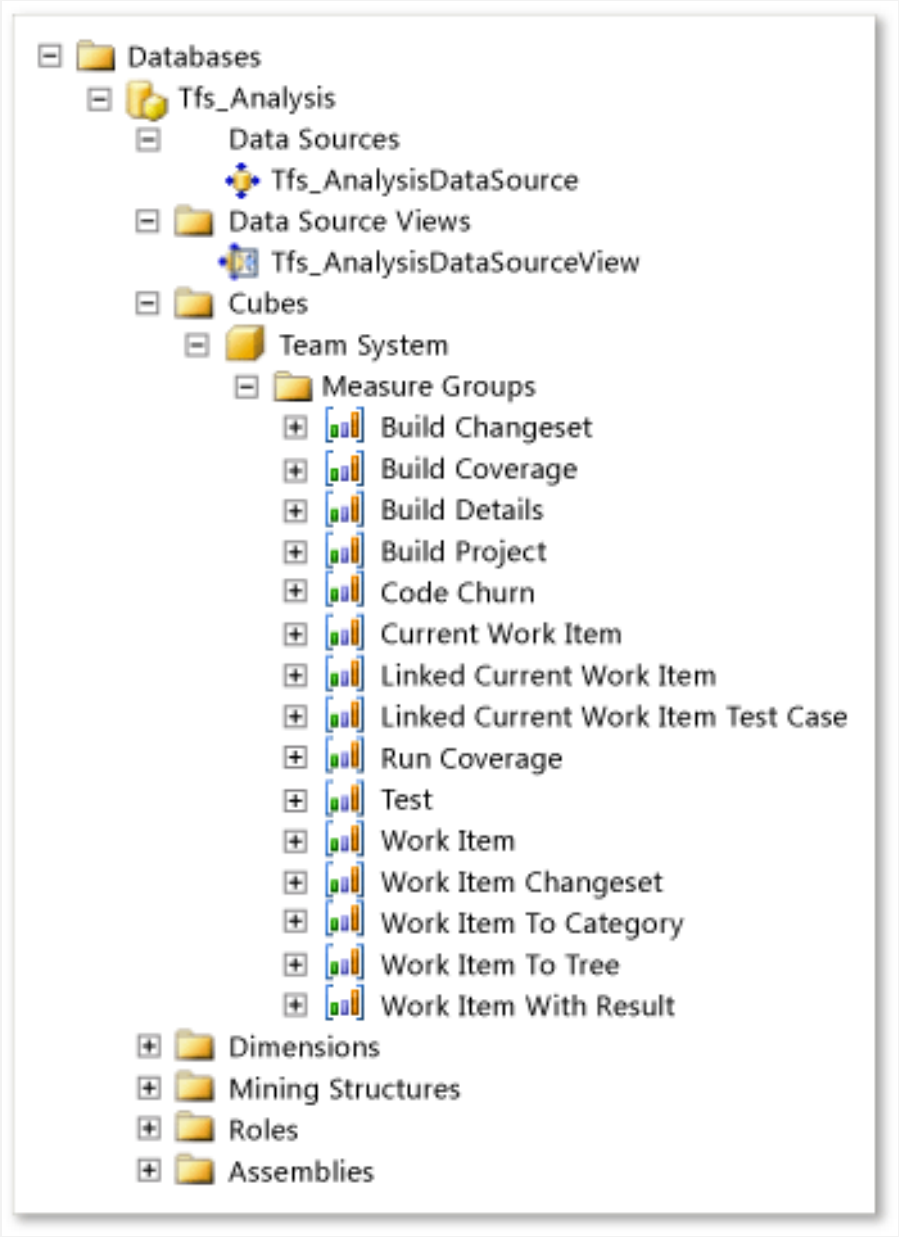
Visual Studio

Test result	Description
Passed	Number of test cases that passed.
Failed	Number of test cases that failed.
Inconclusive	Number of test cases that ran and were assigned a status of Inconclusive.
Blocked	Number of test cases that are currently blocked from running.
Never Run	Number of test cases that have never been run.
Other	Number of test cases that were run and assigned one of the following states: Aborted, Not Executed, PassedButRunAborted, NotRunnable, Disconnected, Timeout, Pending, In Progress, or Inconclusive.

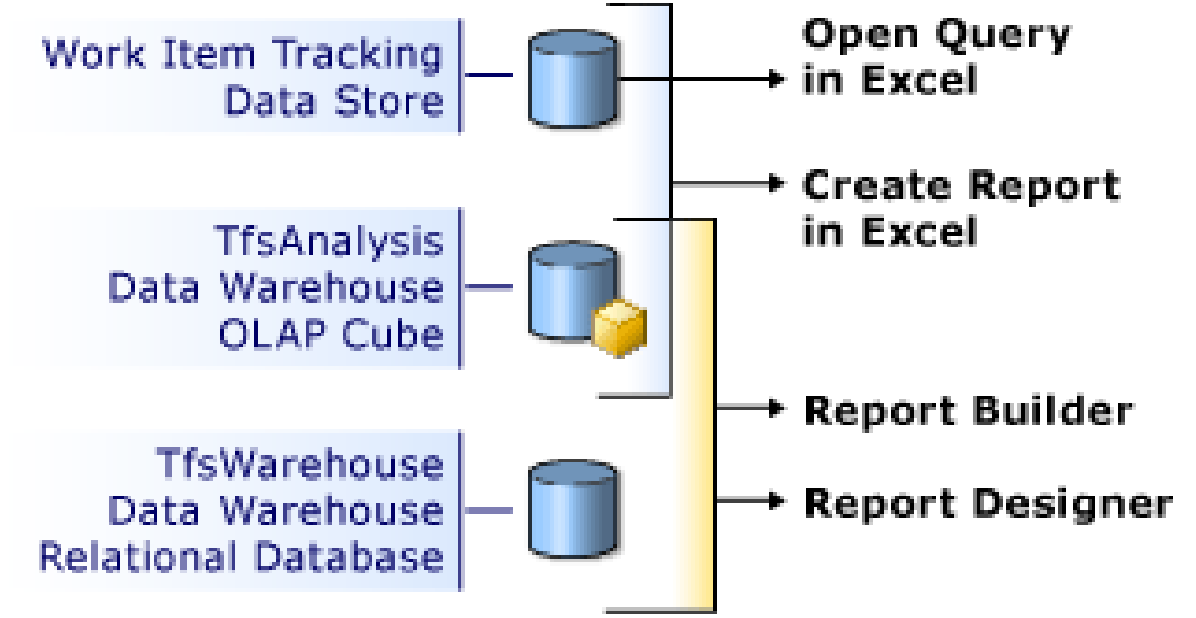
Reporting in TFS



Visual Studio


Databases
Tfs_Analysis
Data Sources
Tfs_AnalysisDataSource
Data Source Views
Tfs_AnalysisDataSourceView
Cubes
Team System
Measure Groups
Build Changeset
Build Coverage
Build Details
Build Project
Code Churn
Current Work Item
Linked Current Work Item
Linked Current Work Item Test Case
Run Coverage
Test
Work Item
Work Item Changeset
Work Item To Category
Work Item To Tree
Work Item With Result
Dimensions
Mining Structures
Roles
Assemblies

Reporting Tools



Excel Reports

Good choice to create reports for:

- Self-service analysis
- Sharing via a portal or dashboard
- Integrating data from disparate sources

Reporting Services Reports

Good choice to create reports that are:

- Consistent
- Scalable
- Manageable
- Fixed-format
- Support automatic distribution

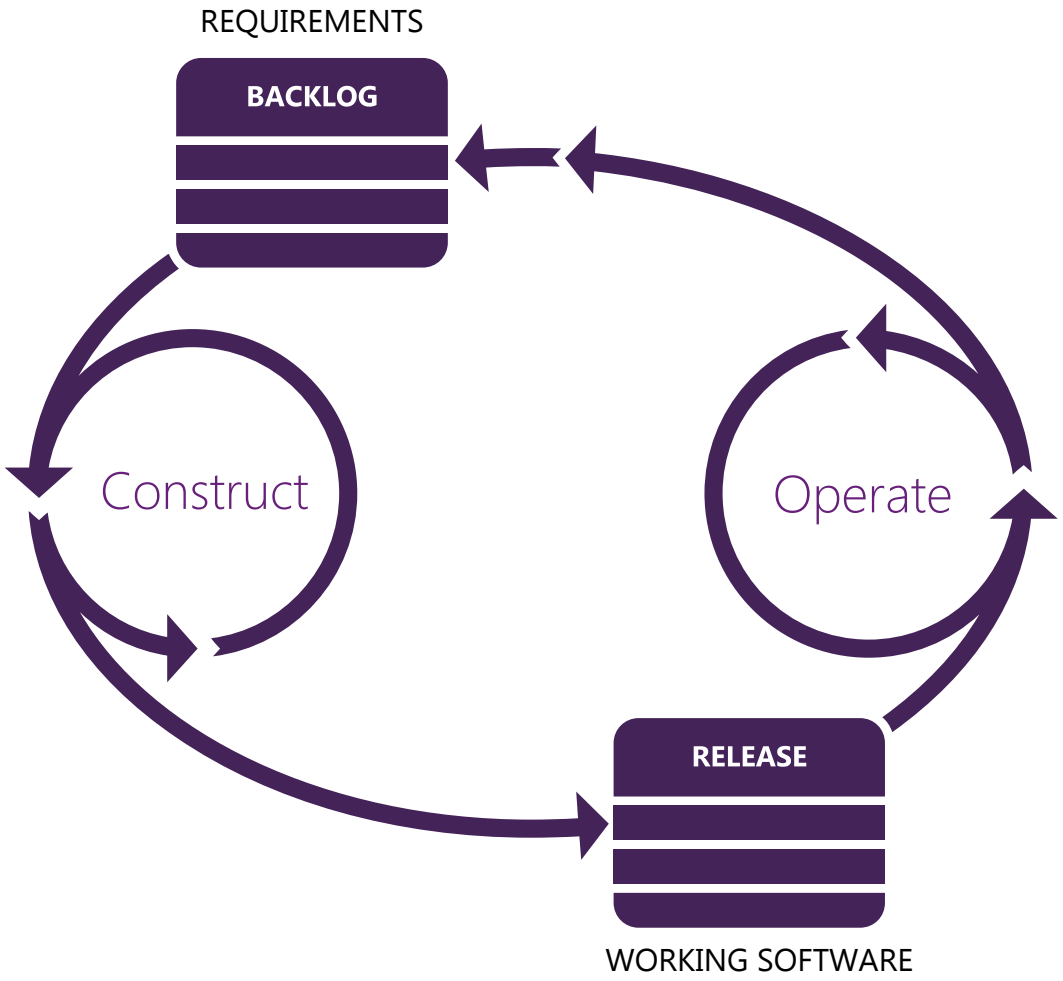
Built in SSRS reports

- Demo

Operation

Continuous value

Solutions



Plan

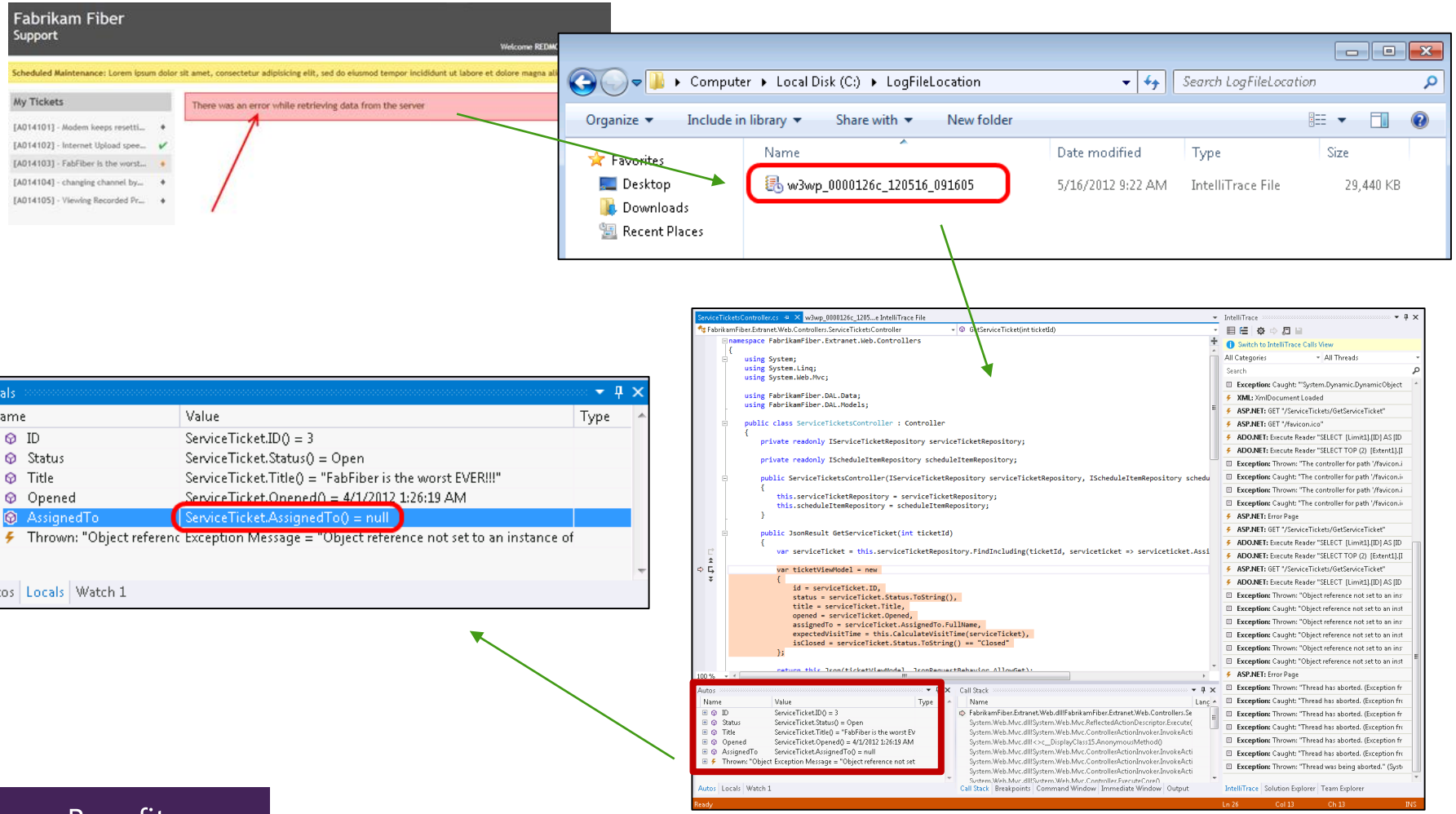
Develop

Release

Operate

- Team Foundation Server and System Center integration
- IntelliTrace in production for actionable diagnostics

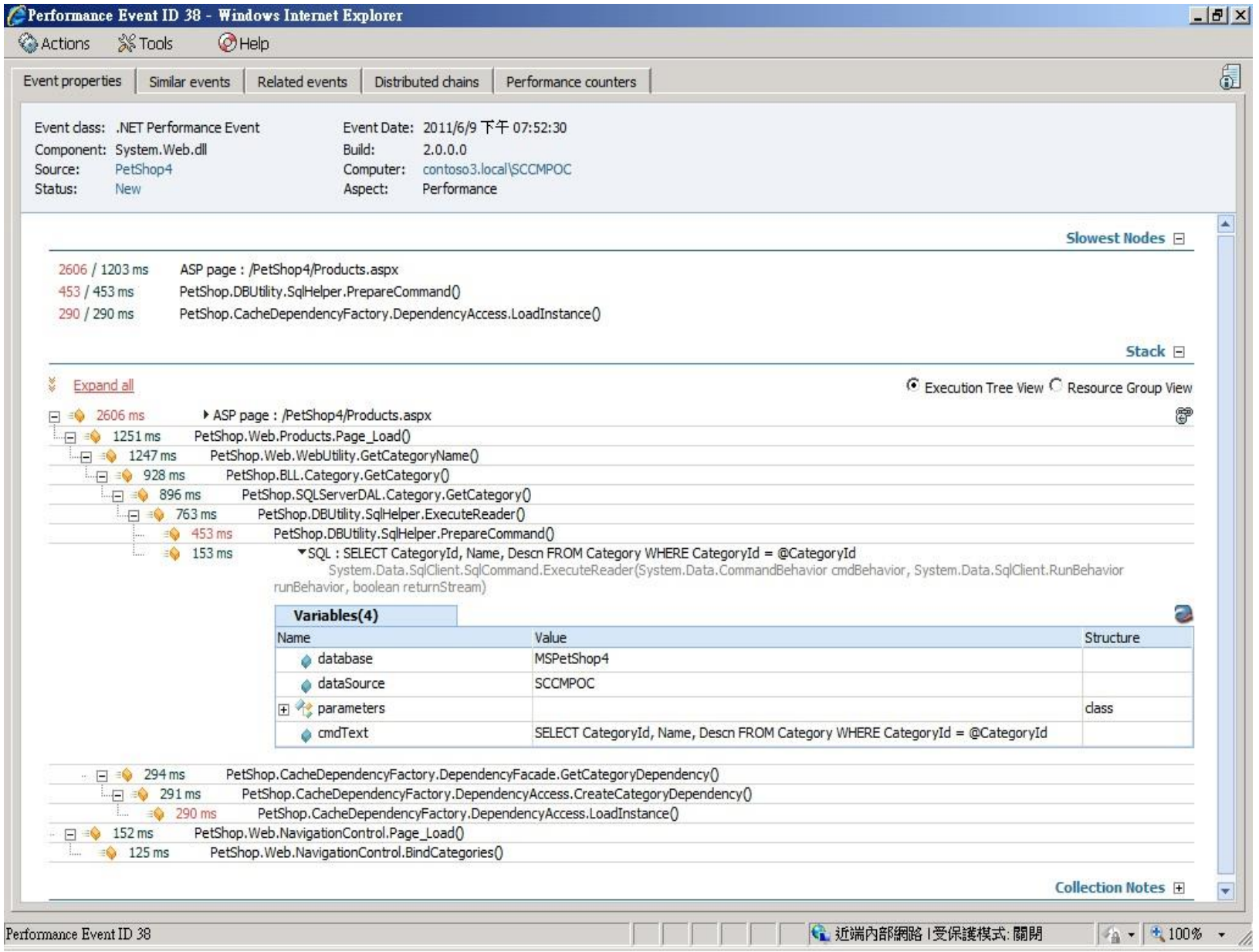
Intellitrace



Benefits

Live trace and log capability on production issues

Intellitrace – Performance Alert (New in VS2013)



Summary

MSDN Benefits

- Unlimited Servers and Tools for Testing
  - **Windows Server, SQL Server – All Editions (i.e.: SQL Enterprise)**
  - **SharePoint Server, Dynamics, BizTalk (Premium & Ultimate)**
- Access to Latest & Legacy Software downloads (Visual Studio 2013, Visual SourceSafe, VB6)
- Concurrent Use (VS2013, VS2008, VB6)
- Office 2013 Professional Plus (Premium & Ultimate)
- Technical Support Incidents
- Online Courses, MSDN Magazines, Windows Store Developer Accounts, O365 developer account (Premium & Ultimate), etc…
- Team Foundation Server (Pro, Test Pro, Prem, Ultimate), 1 Server License and 1 Client CAL

The new era of Cloud Development

Visual Studio

Windows Azure

msdn

MSDN Azure credits

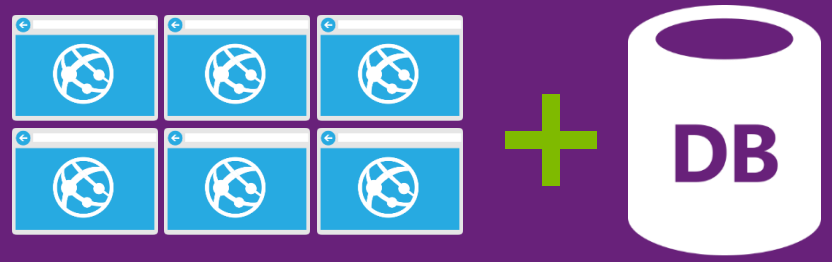
$50/month
for Visual Studio Professional / Test Professional

$100/month
for Visual Studio Premium / MSDN Platforms

$150/month
for Visual Studio Ultimate

3 VMs for 16 hours a day

80 VMs for 20 hour load test



Up to 500 web sites plus SQL DB

Microsoft

© 2013 Microsoft Corporation. All rights reserved. Microsoft, Windows, Windows Vista and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries.
The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.