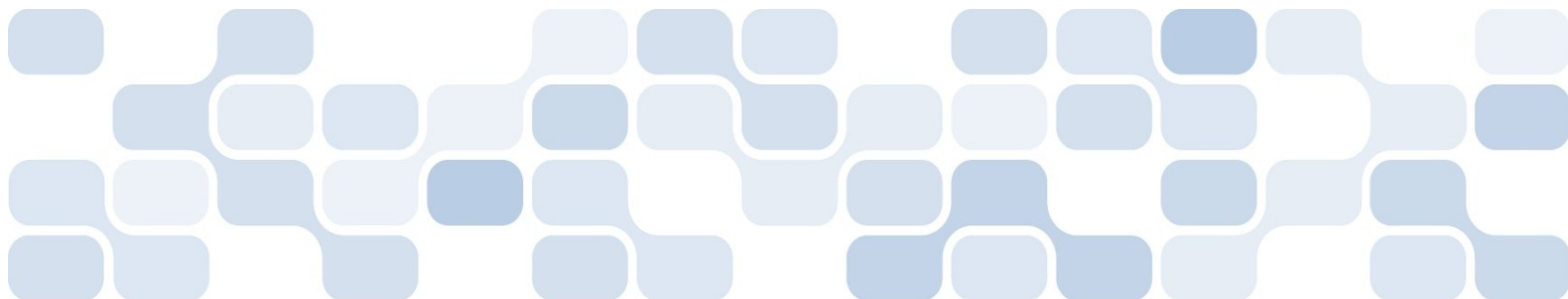


Key Benefits of Microsoft[®] Visual Studio[®] Team System

White Paper

November 2007

For the latest information, please see www.microsoft.com/vstudio



The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. 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 presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form, by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, Excel, MSDN, SharePoint, SQL Server, Visual SourceSafe, Visual Studio, and the Visual Studio logo are trademarks of the Microsoft group of companies.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

CONTENTS

Introduction.....	1
Collaborate and Communicate	2
Drive Predictability	3
Manage Team Workflow	4
Use Familiar Tools.....	5
Ensure Quality, Early and Often	6
Integrate Work Frequently	7
Make Real-time Decisions	8
Summary	9

INTRODUCTION

Software development teams of all sizes struggle with many of the same issues, such as poor communication and collaboration, a lack of consistent processes, and a scramble to ensure quality at the end of the development cycle. And although most teams struggle with the overhead of information gathering and reporting, they still suffer from an inability to make accurate, real-time decisions.

Microsoft Visual Studio Team System is an integrated application lifecycle management (ALM) solution comprising tools, processes, and guidance. With Visual Studio Team System, all members of a software development team can:

- Collaborate and communicate more effectively with other team members and business stakeholders.
- Ensure quality throughout all phases of the development process.
- Improve visibility into project status and quality, and use that data to make informed, real-time decisions.

The remainder of this document examines the key capabilities of Visual Studio Team System that contribute to the above benefits.

COLLABORATE AND COMMUNICATE

Most software development teams use a number of stand-alone tools to manage the application lifecycle. Specifications and requirements are stored on file shares or SharePoint® sites; tasks assigned to developers are managed using e-mail; bugs are tracked with spreadsheets; and source code resides in one or more version-control systems. With important information in so many different places, it's hard for team members to effectively work together, resulting in additional effort and the potential for miscommunication.

Integrated Team Server

Visual Studio Team System facilitates teamwide communication and collaboration by providing a unified repository for all project data, along with the tools to define, enforce, and automate desired processes. At its heart is Visual Studio Team System 2008 Team Foundation Server, which supports source code and version control, work item tracking, automated builds and quality checks, and more. Configurable process templates tie all team activity together, with changes in status gathered automatically as part of everyday workflows. Automatic notifications keep team members informed of key events, such as a failed build. And with all information in one place and tied together, all efforts are traceable back to initial requirements and all stakeholders have deep, real-time visibility into project status and quality.

Support for All Disciplines

Team System Team Foundation Server was designed to meet the needs of all members of the extended development team, not just software developers. Business analysts and architects can manage scenarios, requirements, and design documents; project managers and development leads can assign, track, and report on project status; database professionals can manage database schemas and deployment scripts; and testers can manage regression and load tests, determine test coverage, and keep a close eye on quality trends. The Visual Studio Team Explorer client provides all team members with a single point of access to all project artifacts and data, while a prebuilt, customizable SharePoint portal extends the same information to remote team members and other project stakeholders.

Key Benefits

By bringing all project artifacts, data, and status together in one place, Visual Studio Team System enables more effective team communication and collaboration. Defined and consistently enforced processes, full traceability, and built-in status reporting help maximize individual productivity, at the same time helping to ensure that the efforts of each team member remain well defined and aligned with the team's overall priorities.

DRIVE PREDICTABILITY

Most development teams recognize the value of consistent processes. However, they may not have such processes, know what they need, or know how to get started. Teams that have established processes may find that the processes are not consistently followed, or that adherence requires significant manual effort. Few teams have matured to the point that they can consistently and effortlessly follow established processes and, even if they have, must still deal with issues such as reorganization, new regulatory or compliance requirements, and bringing new team members up to speed.

Defined and Consistently Followed Processes

Visual Studio Team System includes integrated process templates to help teams deliver predictable results, continuously improve and adapt, and more effectively collaborate and communicate. The process templates are fully integrated with other Visual Studio Team System features, such as work item tracking and source code control, providing both the ability to define desired processes and the means for all team members to productively work within those processes.

Visual Studio Team System includes two out-of-the-box process templates: Microsoft Solutions Framework for Agile Software Development and Microsoft Solutions Framework for CMMI (Capability Maturity Model Integration) Process Improvement. Each provides a set of predefined work items, HTML-based process guidance, a SharePoint portal, and a set of predefined reports. Templates for additional methodologies such as Scrum are available in the [Visual Studio Team System Developer Center](#) on MSDN®, the Microsoft Developer Network.

Configurable and Adaptable Templates

Although prebuilt process templates can provide a strong head start, they're not hard-coded. Teams can tailor them to support their own processes or can define new templates from scratch. Either way, the Process Template Editor makes it easy to create new types of work items, specify content for a work item type, define which work items are automatically generated for a new project, and define the relationships between work item types. Teams also can customize process guidance, SharePoint portal layout, and reports.

Key Benefits

Visual Studio Team System can help development teams define, adopt, and enforce consistent processes. In turn, that improved predictability can help increase project velocity, accelerate development cycles, and facilitate more accurate estimation of those cycles, thereby helping teams to deliver results better, faster, and more consistently.

MANAGE TEAM WORKFLOW

Development teams deal with many different types of work items, such as scenarios, use cases, requirements, developer tasks, change requests, test passes, and bugs. However, most teams lack a good way to manage all those work items, define the relationships between them, and trace those relationships and the effects of status changes throughout the application lifecycle. Through such capabilities, teams can better gauge progress toward goals and ensure that resources are not being wasted on unnecessary work.

Detailed Work Plans

Work item tracking in Visual Studio Team System provides an efficient way to manage and monitor the status of all project-related activity. All projects have detailed work plans, with initial work items generated automatically based on process templates. Each work item typically has a title; a description; a team member to whom the work is assigned; and a current state, such as proposed, active, resolved, or closed. Work items can also have links to other work items, attachments, and custom fields.

Full Visibility and Traceability

With relationships between work items clearly defined and changes in the status of work items collected automatically, a team's progress-to-goals remains visible at all times, even as work is handed off among team members. For example, a business analyst may break down scenarios into requirements, which a development lead divides into tasks for developers. As developers finish the tasks, source code-control policies require them to associate their checked-in code with work items. Because those change sets are associated with each build, as a new build passes QA tests, all stakeholders can see that, for example, 60 percent of the work required to meet all project requirements is now complete.

Key Benefits

Work item tracking provides an efficient way to manage the efforts of all team members, with full traceability back to initial project goals and real-time visibility into team progress. Team members will know how their assigned tasks are related to project goals as a whole, and project managers and leads can ensure that no unnecessary work is assigned. And because all changes to work items are logged and fully auditable, those same capabilities can help address complex compliance and regulatory requirements.

USE FAMILIAR TOOLS

Many development teams want an integrated solution for application lifecycle management but also want to avoid the lost productivity associated with having to learn new tools.

Make Use of Existing Tools and Skill Sets

Visual Studio Team System facilitates adoption by enabling team members to use tools that are already familiar to them. Project managers can design an itemized list of work items, assign those work items to developers, and track their completion using Microsoft Office Project Professional, or they can use Microsoft Office Excel® spreadsheet software to make global changes and quickly reassess priorities. Architects, software developers, development leads, database professionals, and testers can continue to use the Visual Studio integrated development environment, including Visual Studio 2008 Professional Edition or any of the Visual Studio Team System 2008 Editions. Remote team members and business stakeholders can use Team System Web Access to take advantage of all Visual Studio Team System features.

Key Benefits

Visual Studio Team System helps reduce “soft” adoption costs by integrating with the desktop applications that people already know and use. Instead of having to learn new tools, all team members can continue to use the tools that help them be the most productive.

ENSURE QUALITY, EARLY AND OFTEN

Quality is often neglected during most phases of the development process. Instead, it is measured near the end of the process, when code is handed off to be tested. This approach often results in extra work and a slipping schedule as features thought to be complete are pushed back to developers for rework. Even if code passes initial QA tests, performance and scalability issues can remain undetected until software is deployed into production.

A Quality-Centric Tool Set

Visual Studio Team System provides tools for ensuring quality throughout all phases of the application lifecycle, helping teams to deliver high-quality software faster and with less rework. From tracking requirements during planning to performance and load testing of the completed application, Visual Studio Team System provides the tools needed to ensure all aspects of software quality.

Focus on Quality Throughout the Application Lifecycle

The focus on quality begins during the planning phase, before the first line of code is written, when work item tracking helps thoroughly map scenarios to requirements. As architects design a solution that meets those requirements, tools such as the Application Designer, Distributed System Designer, and Deployment Designer help ensure that the solution will deliver the necessary performance, scalability, and manageability.

During development, before code is checked in, static code-analysis tools help prevent coding errors and potential security issues, while performance profiling and hot-path analysis help avoid potential performance and scalability problems. Developers can easily create unit tests to validate application and database objects, using code coverage analysis tools to determine the completeness of those tests. Tools for examining code metrics can gauge the complexity and maintainability of code.

QA engineers can begin writing test cases and load tests early, mapping them to work items and managing all test activity with Visual Studio Team System. As requirements change, full traceability between work items helps them ensure that test coverage remains complete. Prior to production deployment, comprehensive load-testing tools help QA engineers validate the application's performance and scalability as a whole.

Key Benefits

Visual Studio Team System enables development teams to ensure quality throughout all phases of the application lifecycle, not just near the end. Such an approach helps minimize the rework-and-retest “churn” that typically occurs near the end of each development iteration, during which tradeoffs on quality versus schedules must often be made—in turn helping to increase software quality, reduce time-to-market, and decrease overall development costs.

INTEGRATE WORK FREQUENTLY

Another quality-related area where most teams see room for improvement is source code management and version control. One common issue is the integration of individual developer efforts into the official code base, which can result in the all-too-frequent “broken build”—and force QA resources to sit idle until the issues have been resolved and a new build is ready to test.

Comprehensive Version Control

Visual Studio Team System helps solve these pains by providing a comprehensive, flexible, version-control system. This functionality is not a mere upgrade to Microsoft Visual SourceSafe® version-control system; rather, the version-control capabilities of Team System Team Foundation Server were designed from the ground up to deliver the same scalability, performance, and reliability of its underlying data store, which is based on Microsoft SQL Server™ database software. Built-in tools aid in the migration of source code and change history from Visual SourceSafe and other popular version control software.

Some key version-control features in Team System Team Foundation Server include atomic check-ins, which help maintain the integrity of source-code files, and policies that require developers to perform unit tests or static code analysis prior to checking in code and to associate all checked-in code with work items. Support for “shelving” code enables developers to store work in process on the server without checking it in, and a new “get latest on edit” feature checks for new code on the server when a developer starts to edit a local copy—just in case someone else has made changes since the code was checked out.

Powerful Build Server

Version control in Visual Studio Team System is complemented by its Team Build features, including support for continuous integration builds every time someone checks in code. Unit tests and build verification tests can be run as part of the process, and if the build or tests fail, designated stakeholders are automatically notified. Flexible build definitions also help reduce the time spent managing scripts for “official” builds, providing the ability to easily automate compilation, unit tests, static code analysis, virtual server configuration, Web site or Web services configuration, application installation, database deployment, test-data generation, and load tests.

Key Benefits

The version-control and Team Build features in Visual Studio Team System help teams ensure quality by enabling them to easily and frequently integrate the work of individual team members—and validate that the combined efforts work as expected.

MAKE REAL-TIME DECISIONS

Regardless of the maturity of their processes, many development teams lack full visibility into the application lifecycle—and thus they are unable to easily answer questions such as “Does the application meet all requirements?” and “Are we on schedule?”

Business Intelligence for Development Teams

Powered by an integrated SQL Server-based data warehouse, the business intelligence features in Visual Studio Team System provide the information needed to make informed, real-time decisions with just a click on a link. And because SQL Server Reporting Services is the engine underlying Visual Studio Team System business intelligence, teams can just as easily customize existing reports as they can create new ones.

One highly useful out-of-the-box report is Remaining Work, which provides a comprehensive view of all remaining work items, enabling development leads and project managers to easily see progress, identify bottlenecks, and, if necessary, reallocate resources. Another useful prebuilt report is Quality Indicators, which provides a combined view of unit test success rates, code coverage by unit tests, code churn, and active bugs—all tracked over time.

Key Benefits

Visual Studio Team System provides the information needed to make informed, real-time decisions on iteration planning, test coverage, bug trends, project schedules, resource allocation, and more—in turn helping development teams to deliver on time and on budget, and to ensure that projects meet all requirements. Project managers and development leads can be ready for team meetings in minutes instead of hours, saving everyone the effort of having to manually communicate status, and can answer questions posed by external stakeholders and upper management on-the-fly.

SUMMARY

Microsoft Visual Studio Team System 2008 enables software development teams to work, communicate, and collaborate more productively; enhance predictability and quality throughout the application lifecycle; and make better decisions through real-time visibility into project status and quality. Teams desiring to realize these benefits can adopt Visual Studio Team System gradually, starting with features like version control, work item tracking, and built-in reporting before starting to take advantage of more advanced capabilities like Team Build. Regardless of the order in which new capabilities are deployed, teams that adopt Visual Studio Team System can be assured that it will provide the flexibility to support the way they want to work instead of dictating how they must do so.

For more information on Microsoft Visual Studio Team System, visit the Web site at www.microsoft.com/teamsystem.