

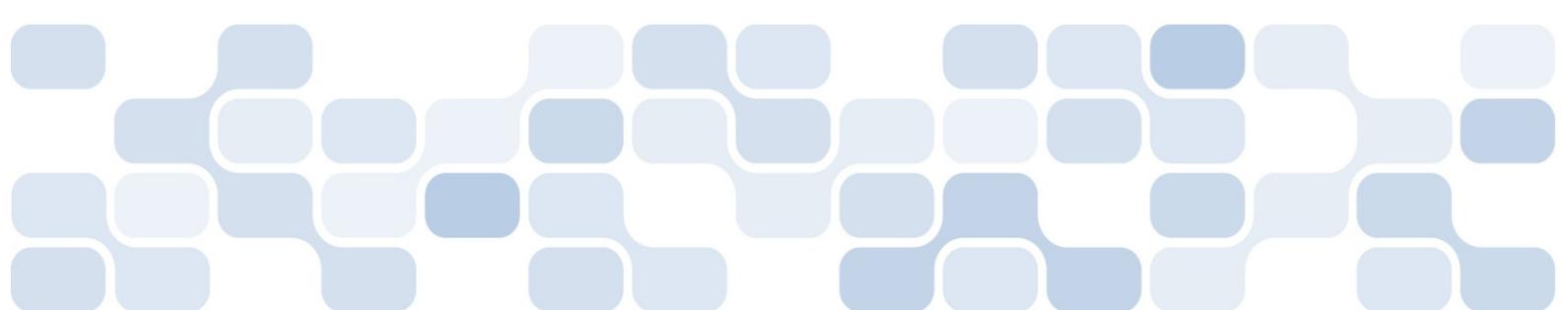


# **Requirements Management and Visual Studio Team System**

White Paper

September 2008

For the latest information, please see [www.microsoft.com/teamsystem](http://www.microsoft.com/teamsystem)



This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

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, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

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 or 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.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, Excel, Outlook, Visio, and Visual Studio are trademarks of the Microsoft group of companies

All other trademarks are property of their respective owners.

# **CONTENTS**

<b>Overview.....</b>	<b>1</b>
<b>Requirements Management and the Requirements Life Cycle .....</b>	<b>2</b>
<b>What Are Requirements? .....</b>	<b>5</b>
<b>Team System Road Map of the Requirements Life Cycle .....</b>	<b>6</b>
<b>Requirements Activities and Team System Solutions .....</b>	<b>7</b>
<b>Requirements Futures with Team System 2010.....</b>	<b>21</b>
<b>Requirement Partners .....</b>	<b>23</b>
<b>Appendix A: Work Breakdown in Office Excel.....</b>	<b>28</b>
<b>Appendix B: Requirements-Quality Report .....</b>	<b>29</b>
<b>Appendix C: Requirements-Progress Report.....</b>	<b>30</b>

---

## **OVERVIEW**

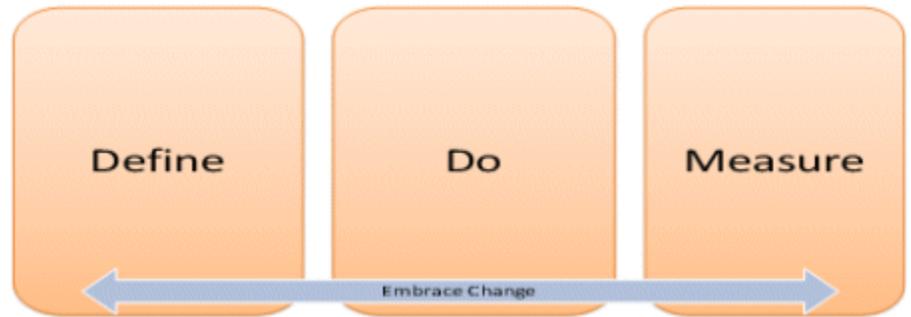
Failures to define, communicate, or validate business requirements are the root causes of many software-project delays, overruns, and failures. There is a growing realization in the industry of the importance of proper development of requirements from every level in the organization. Microsoft Visual Studio Team System, along with Visual Studio partner solutions, can help improve the requirements-development process.

The goal of this white paper is to outline how to use Visual Studio Team System for successful requirements management by using Visual Studio Team System 2005 and Visual Studio Team System 2008. In addition, you can use this paper to learn about some of the problems that Microsoft intends to address in the next release: Visual Studio Team System 2010.

---

## REQUIREMENTS MANAGEMENT AND THE REQUIREMENTS LIFE CYCLE

Requirements through the application life cycle can be defined quite simply, as shown in Figure 1.



**Figure 1. Requirements life cycle**

In simple form, firstly define the requirements. Elicit the right requirements from customers, translate them into solutions, prioritize, refine, break down, and plan the work and tests.

Then, execute against the requirements. Write code, write tests, check in, build, and track both team progress and requirements quality.

Lastly, deploy into production or release the solution, and measure the KPIs, SLAs, and customer feedback to elicit the next set of requirements.

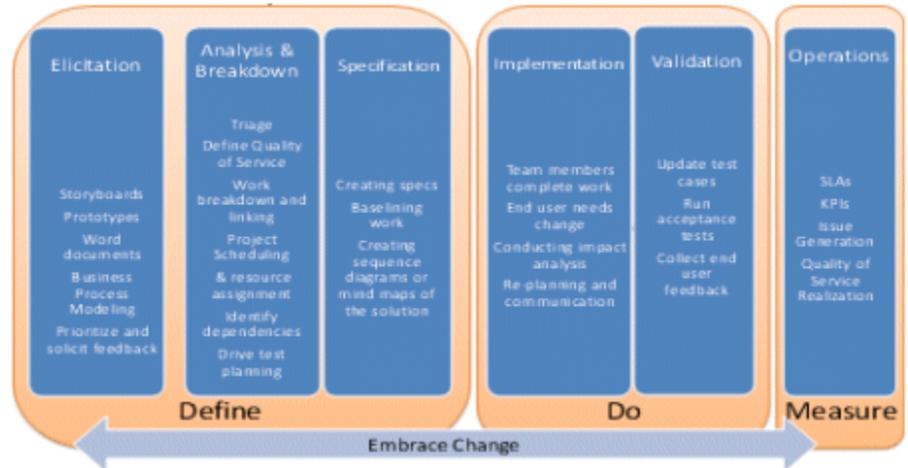
Throughout this process, change occurs. Requirements change, work breakdown changes, new work is identified, work takes longer than expected, scoping occurs, people leave or join the team, and so on. Change cannot be avoided, so requirements activities cannot be considered one-time activities or strictly sequential. The requirements life cycle must embrace change throughout, as life-cycle phases iterate and bleed into each other, and change is able to ripple throughout all phases and activities.

Traceability is discussed often as the solution to embracing change; it is one aspect of change management that must occur at all phases and all activities of the requirements life cycle. Traceability relates work across activities in a meaningful way and usually is represented through link relationships in Application Life-Cycle Management (ALM) tools. To enable traceability, tools that are used at each phase of requirements management and throughout the application life cycle must be integrated in order to keep an accurate, up-to-date reflection on related work and artifacts.

Another aspect of embracing change is frequent customer connection throughout the requirements life cycle. As customer needs change and the solution evolves, it makes sense to validate regularly that the solution meets needs. The most successful projects communicate regularly with customers to stay in sync on changes in both work and needs.

The Project Management Body of Knowledge Guide (PMBOK) and several other sources have defined more granularly the activities that are related to

requirements. These activities can be mapped over the requirements life cycle, as shown in Figure 2.



**Figure 2. Requirements activities**

Not all of these activities are necessary for every project or every organization. For many, this segmentation might prove too fine-grained. The requirements-management process should be able to “stretch to fit” the diverse needs of different teams or projects. It is important for each activity to embrace change and integrate with the other activities to provide a continuous requirements life cycle that supports the application life cycle.

Team System is poised as the integrated platform on which to base the requirements life cycle. Integrated tools that enable the team to operate on common data, the ability to define the level of ceremony in your process, and a platform to tie it all together provide a level of visibility and change management throughout the requirements and application life cycle.

### **Requirements Management versus Requirements Elicitation**

Many people equate the full requirements life cycle with requirements management. However, we are seeing more people think about requirements elicitation as distinct from requirements management. This differentiation maps fairly well to the road map on which Team System will execute on providing solutions to address the requirements life cycle.

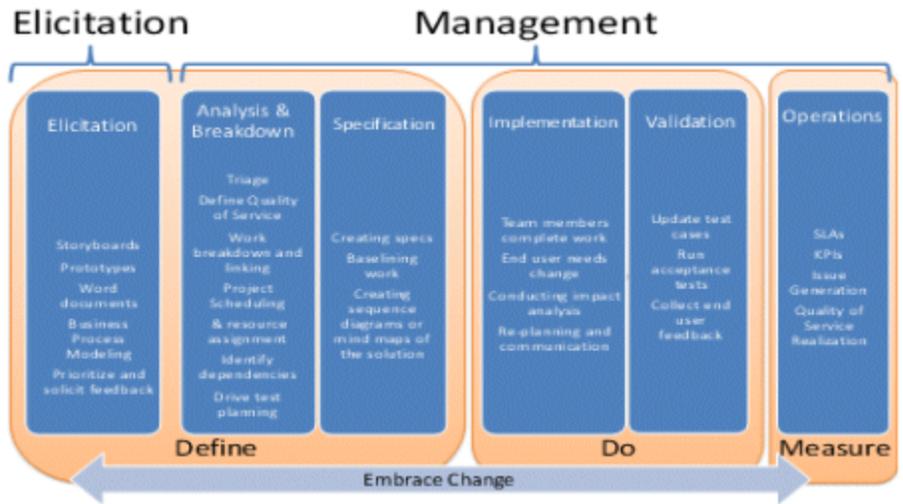


Figure 3. Elicitation versus management

---

## WHAT ARE REQUIREMENTS?

Requirement is the industry term for describing end-user need or the level of service that is expected from an application. Requirements tend to vary in granularity and meaning. In typical businesses, requirements can be categorized as:

- **Business requirements.** These describe the overall objectives of the system or how it will affect the business.
- **User requirements.** These describe user needs and problems.
- **Functional requirements.** These describe specific behavior of the system and the internal workings of the software.
- **Quality of Service requirements.** These apply to the whole system and are the “abilities,” such as localizability, accessibility, performance, scalability, and so on.

The first step in using Team System to manage requirements is to create a requirement work-item type, which enables the requirement to be tracked and traced through the development process.

Business analysts in many organizations are formally responsible for interfacing with the end user; eliciting, refining, analyzing, and validating the requirements; and reporting on requirements progress. However, many roles—such as developers, testers, or project managers—might be involved or responsible for collecting requirements. Regardless of how your software-engineering process defines requirements or who collects them, Visual Studio Team System Team Foundation Server helps you capture and store requirements when you create requirement work-item types.

---

## TEAM SYSTEM ROAD MAP OF THE REQUIREMENTS LIFE CYCLE

Team System provides support for several of the activities in the requirements life cycle, as well as a rich ecosystem for partners to plug in and provide additional activities that can be well integrated. The goal is for these activities to be integrated and enacted on the Team System platform, as your process and your process template dictate.

Team System began by addressing issues in the implementation phase. Visual Studio continues to be an established productivity tool for developers, and Team System integrates with Visual Studio to enact the process by which developers and testers complete their work. The strategy in addressing the requirements life cycle is to start with the implementation phase and branch out from there over the course of several releases. We'll begin by talking about what is possible with Visual Studio Team System 2005 and Visual Studio Team System 2008, and then discuss in detail the road map and partner products.

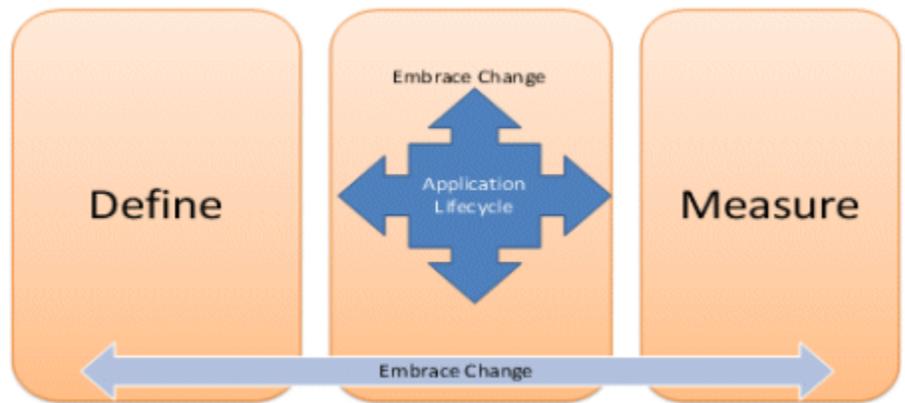


Figure 4. Microsoft requirements life-cycle road map

## REQUIREMENTS ACTIVITIES AND TEAM SYSTEM SOLUTIONS

This section discusses in detail each of the requirements activities, and how to work with released Team System and partner products to elicit, refine, specify, implement, and validate requirements—tracing them through the application life cycle.

### Requirements Elicitation and Refinement

*Requirements elicitation and refinement* is the act of collecting requirements from the end user. Successful projects communicate regularly with their end users—eliciting requirements throughout the project life cycle. The bulk of the requirements-elicitation activity and refinement usually peaks early in the life cycle of a project, but continual communication with end users generates the highest customer satisfaction with the end product. Requirements elicitation involves noting the customer needs quickly, describing solution ideas to customers, agreeing on priority, agreeing on timeline, and iterating on solution ideas. Requirements are added, removed, or changed throughout this process, which is challenging for several reasons, including the following:

- **End users/Customers have a hard time describing what they need.** “I don’t know what it looks like; however, if you show it to me, I can tell you if it’s right.”
- **What end users describe almost always is incomplete.** End users are not product designers, and their suggested solutions almost never describe the full set of needs. Someone must think the design through thoroughly.
- **Work transitions from several different tools.** There are several different tools and techniques to elicit requirements. As the elicitation process progresses, requirements can be scattered across several different tools and it is difficult to get one view of the priority and state of the requirements.

Eliciting requirements requires both a variety of tools that are based on personal preference and changing tools appropriately, as the elicitation process progresses.

Activity	Purpose/Timeframe	Appropriate tools	Partner tools
<b>Take early notes.</b>	Be able to jot down needs quickly as the customer describes them.	Text-based tools, such as Microsoft Office Word	Personify Design TeamSpec  Borland CaliberRM  Techno Solutions TopTeam Analyst

<b>Activity</b>	<b>Purpose/Timeframe</b>	<b>Appropriate tools</b>	<b>Partner tools</b>
<b>Define a high-level vision.</b>	Describe the purpose of the project and the key success criteria.	Text-based tools, such as Office Word	Personify Design TeamSpec  Borland CaliberRM  Techno Solutions TopTeam Analyst
<b>Relate requirements.</b>	Help think about the problem or the solution by grouping related requirements.	Hierarchy in a database or spreadsheet  Mind mapping	Borland CaliberRM  Techno Solutions TopTeam Analyst Mindjet RM
<b>View all requirements.</b>	Form an understanding of the requirements by viewing a list and/or the relationships between them.	Spreadsheets  Graphs of relationships  Mind mapping	Mindjet RM  Ravenflow RAVEN  Borland CaliberRM  Techno Solutions TopTeam Analyst  Compuware OptimalTrace stpsoft stpBA
<b>Triage.</b>	View a list of all of the requirements, and make adjustments to priority, descriptions, and so on.	Spreadsheets	

<b>Activity</b>	<b>Purpose/Timeframe</b>	<b>Appropriate tools</b>	<b>Partner tools</b>
<b>Define how the user uses the solution.</b>	Describe the experience through user steps.  Note the actors and the actions that are taken throughout the solution.	Text-based user stories or scenarios in word-processing tools  Activity diagrams in UML  Sequence diagrams in UML	Personify Design TeamSpec  Borland CaliberRM  Techno Solutions TopTeam Analyst  Ravenflow RAVEN  Compuware OptimalTrace  TechnoSolutions Visual UseCase
<b>Define the user experience.</b>	Describe the solution in terms of the visual customer experience.	Storyboarding tools  Prototypes	Compuware OptimalTrace  stpsoft stpBA
<b>Create a contract.</b>	Record an agreed-upon set of requirements.	Text-based tools, such as Office Word	Personify Design TeamSpec  Borland CaliberRM  Techno Solutions TopTeam Analyst
<b>Model business processes.</b>	Lay out the flow of business logic, events, and rules for the application.	Modeling tools	stpsoft stpBA  TechnoSolutions TopTeam Analyst/Visual UseCase  Holocentric Modeler
<b>Define the technical solution.</b>	Determine the technical requirements of the solution and how it will be built.	Prototypes  Data-flow sequence diagrams  Mind maps	stpsoft stpBA  Mindjet RM  Ravenflow RAVEN  TechnoSolutions Visual UseCase

Activity	Purpose/Timeframe	Appropriate tools	Partner tools
<b>Use end-user feedback.</b>	Solicit feedback from customers on the proposed solution.	Voting tools  Comments on the solutions in the tools  Reports on bugs, feedback	

Visual Studio Team System 2005 and Visual Studio Team System 2008 help drive requirements through the life cycle as soon as they have been entered into the system as work items. There are a few options with requirements elicitation. One option is to elicit requirements in spreadsheets, which are entered into Team Foundation Server through the Microsoft Office Excel Integration functionality. Another option is to enter requirements directly in the Team Foundation Server requirement work-item forms in Team Explorer. Entering this information helps provide a view of all requirements, helps triage requirements, and integrates with the rest of the application life cycle.

Additionally, several partner solutions are available to assist with both requirements elicitation and the entering of those requirements into Team Foundation Server. For more information, please see the “Requirements Partners” section at the end of this white paper.

Lastly, the Visual Studio Tools for Office 2005 Requirements Authoring Starter Kit provides a base level of integration with Office Word and Team Foundation Server. This basic tool demonstrates the integration of text with Team Foundation Server.

### Requirements Analysis and Breakdown

Requirements breakdown is the act of translating the customer problems and needs into solutions and work. Requirements are categorized into either Functional or Quality of Service; they are broken down hierarchically, and sub-requirements are created. Tasks are derived; dependencies and acceptance tests are identified. The challenges in refining requirements include the following:

- **End users think about the end product (the application); development teams think about the intermediate steps to build it.** Software end users do not describe their requirements in terms of features, components, or other artifacts that are easily digestible by the development team, and they need help with translating what they want into engineering plans.
- **The development team has to iterate with the customer to get the most cost-effective plan.** The development team must interpret and cost the requirements. Then, they must bring them to the end user with alternatives. End users must be walked through the solution to understand the cost and design trade-offs; they should

provide feedback into the solution. It is challenging to describe effectively the solution and help end users understand the cost of design trade-offs.

- **Work transitions from several different tools.** As requirements are collected in several different tools, the development team can't be sure that it has the complete list of requirements. The team relies on the business analyst to provide this assurance.

Activity	Purpose/Timeframe	Appropriate tools	Partner tools
<b>View all requirements.</b>	Form an understanding of the requirements by viewing a list and/or the relationships between them.	Spreadsheets Graphs of relationships Mind mapping	Mindjet RM Ravenflow RAVEN Borland CaliberRM Techno Solutions TopTeam Analyst Compuware OptimalTrace stpsoft stpBA Personify Design TeamLook, TeamSpec
<b>Triage.</b>	View a list of all of the requirements, and make adjustments to priority, descriptions, category, and so on.	Spreadsheets	

<b>Activity</b>	<b>Purpose/Timeframe</b>	<b>Appropriate tools</b>	<b>Partner tools</b>
<b>Break-down work.</b>	Help think about the problem or the solution by grouping related requirements and creating sub-requirements or tasks.	Hierarchy in a database Hierarchy in a spreadsheet Hierarchy in a project plan Hierarchy in Office Word Mind mapping	Personify Design TeamSpec Mindjet RM Ravenflow RAVEN Borland CaliberRM Techno Solutions TopTeam Analyst Compuware OptimalTrace stpsoft stpBA
<b>Identify dependencies.</b>	Identify shared work and order work.	Graphs of relationships Mind mapping Links in project plans	Personify Design TeamSpec Mindjet RM Ravenflow RAVEN Borland CaliberRM Techno Solutions TopTeam Analyst Compuware OptimalTrace stpsoft stpBA
<b>Assign resources.</b>	Project managers and leads determine the right people to complete the work.	Spreadsheets Project plans Custom scheduling applications	

Activity	Purpose/Timeframe	Appropriate tools	Partner tools
<b>Schedule work.</b>	Project managers and leads determine cost estimates, timelines, and project schedules.	Spreadsheets Project plans Custom scheduling applications	
<b>Drive test planning.</b>	Link test cases from the test plan to requirements, to ensure that all requirements have test coverage.	Spreadsheets Test-management systems	
<b>Determine acceptance tests.</b>	Enumerate and define tests to validate the user functionality.	Spreadsheets Manual-testing tools	

Visual Studio Team System 2005 and Visual Studio Team System 2008 provide integration between the data that is stored in Team Foundation Server and spreadsheets or project plans. As data is entered as requirement work items, this integration helps connect the development team with the requirements. Both business analysts and development teams can pull up lists of requirements and view all current requirements; they can triage the requirements, and can make sure that those requirements are up to date with customer priorities.

Requirements can be broken down in either Office Excel or Office Project with Visual Studio Team System 2005 and Visual Studio Team System 2008. With Office Excel, requirement titles can be formatted differently in order to indicate child requirements. You can switch the work-item type to indicate a breakdown into tasks. The link information remains in your .xls file, and none of it is persisted in Team Foundation Server. In order to understand work relationships and get an up-to-date view of work breakdown, reopen the .xls file and refresh the data. Appendix A contains an example of work breakdown that is conducted in Office Excel in this fashion.

More commonly, work breakdown is conducted in Office Project by indenting work items. Once again, link information is persisted in the .mpp file, and it is not entered into Team Foundation Server.

Dependency management is conducted in a similar manner with Visual Studio Team System 2005 and Visual Studio Team System 2008. Data that is stored in Office Project or Office Excel to indicate dependencies between work items is preserved only in the .xls or .mpp file, respectively, unless a “related to” link is entered manually in Team Foundation Server. In order to

generate reports out of the Team System reporting warehouse, the manual links must be entered into Team Foundation Server. Up-to-date dependency information is understood by returning to the respective .xls or .mpp file and refreshing the data.

Office Project and Office Excel are the tools of choice to assign resources and schedule work with Visual Studio Team System 2005 and Visual Studio Team System 2008. Work items also can be assigned directly in the Team Foundation Server client.

To make sure that requirements are tested, test results must be associated with requirements. You can accomplish this directly from either a test run or the work-item form itself. As soon as this is complete, reports can be generated to show whether all requirements have tests, as well as the results of those tests.

Appendix B shows an example requirements-coverage report, and Appendix C illustrates a requirements-progress report. We hope to make this process much smoother in future releases of Team System.

### Requirements Specification

The process of describing the solution is called requirements specification. This process involves many of the same tools and activities that are described in the “Requirements Elicitation and Refinement” section. Requirements specification emphasizes the reaching of an agreement or baseline solution from which to move forward.

Activity	Purpose/Time frame	Appropriate tools	Partner tools
Define how the user uses the solution.	Describe the experience through user steps.  Note the actors and the actions that are taken throughout the solution.	Text-based user stories or scenarios in word-processing tools  Activity diagrams in UML  Sequence diagrams in UML	Personify Design TeamSpec  Mindjet RM  Ravenflow RAVEN  Borland CaliberRM  Techno Solutions TopTeam Analyst  Compuware OptimalTrace  stpsoft stpBA

Activity	Purpose/Time frame	Appropriate tools	Partner tools
Specify the solution.	Describe the solution.	Text-based tools, such as Office Word	Personify Design TeamSpec  Ravenflow RAVEN  Borland CaliberRM  Techno Solutions TopTeam Analyst  Compuware OptimalTrace
Create a contract.	Record an agreed-upon set of requirements.	Text-based tools, such as Office Word	Personify Design TeamSpec  Ravenflow RAVEN  Borland CaliberRM  Techno Solutions TopTeam Analyst  Compuware OptimalTrace

Activity	Purpose/Time frame	Appropriate tools	Partner tools
Define the technical solution.	Determine the technical requirements of the solution and how it will be built.	Prototypes Data-flow sequence diagrams Mind maps	Personify Design TeamSpec Mindjet RM Ravenflow RAVEN Borland CaliberRM Techno Solutions TopTeam Analyst Compuware OptimalTrace stpsoft stpBA
Baseline the solution.	Take a snapshot of the requirements and the proposed solution at a given point in time.	Database snapshots and versioning	

Visual Studio Team System 2005 and Visual Studio Team System 2008 provide integration with Office SharePoint document libraries in Visual Studio Team Explorer. This enables you to navigate your specifications easily without leaving the Visual Studio environment. In addition, you can link to documents or other files specifying requirements from within work items. This provides a base level of traceability from the work to the work descriptions. However, it doesn't provide change management or integration to the text or other content within the files. For deeper integration with Office Word, partners have filled many of these gaps. See the "Requirements Partners" section at the end of this white paper for more information.

### **System Analysis and Design**

Several tools and diagramming surfaces aid in this process to refine requirements into a cohesive architecture and to a code skeleton for implementation. Some teams want requirements to trace through this process, to make sure that the solution meets end-user needs. This white paper does not address this portion of requirements management.

### **Requirements Implementation**

Business analysts report on requirements progress and manage change to requirements throughout the application life cycle. Traceability from

requirements into the work on the team is important in order to manage the impact of changes on both customers and the work. This is done either manually by pulling the team together to assess the impact of a change, or by using requirements-management tools and trolling through link relationships to assess the implications. As work is completed, it is important that the development and test teams create and update appropriate link relationships in order to maintain requirements traceability. This process is a huge challenge for several reasons, including the following:

- **Requirements change, and the development team must adjust rapidly and correctly.** Even when you start with a complete and clear set of requirements, the project is under constant change: Customers change their needs due to business changes; the development-team capacity changes; the design changes. It is tough to keep all of this synchronized, and chances are that something becomes out of date.
- **The development work changes, but customers are left in the dark.** The work breakdown is in the development team's language and it is difficult for the business analyst to assess and communicate the impact on the customer requirement.
- **Tools that provide traceability are manually intensive and difficult to maintain.** Even if links are carefully constructed between requirements and the related work and tests, assessing a change is a manually intensive process. If links are out of date or meaningless to the person who assesses the change, the investment in creating the links is not worth the cost. Many requirements-management tools are abandoned because of the lack of value that there is in creating and maintaining link relationships easily.

Activity	Purpose/Time frame	Appropriate tools
<b>Complete work.</b>	Update tasks, write code, complete tests, log bugs, and link tasks or requirements to source code.	Integrated work-item-tracking and source-code-control tools
<b>Update requirements.</b>	Update the requirement scope, priority, and so on, based on changing user needs.	Spreadsheets Database entry Entry directly into requirements-elicitation tools

<b>Activity</b>	<b>Purpose/Time frame</b>	<b>Appropriate tools</b>
<b>Update work breakdown.</b>	Assess the impact of a requirement change on work breakdown, schedules, and work assignments.	Hierarchy in a database Hierarchy in a spreadsheet Hierarchy in a project plan Hierarchy in Office Word Mind mapping
<b>Update dependencies.</b>	Assess the impact of requirements change on dependencies.	Graphs of relationships Mind mapping Links in project plans
<b>Communicate change to customers and team members.</b>	Ensure that the tasks on which the development and test teams are working are appropriate.	Project plans Office Excel spreadsheets Work queues

Both Visual Studio Team System 2005 and Visual Studio Team System 2008 provide integration among work-item tracking, source-code control, Office Excel, and Office Project. This integration enables traceability between implementation activities and the work plans. As developers or testers check-in work, they can mark-off task completion. The project plans or spreadsheets that contain those tasks and the schedule can be refreshed easily by using the Team Foundation Server Office Integration functionality. The .xls or .mpp files that contain the work-breakdown structure should be revisited and refreshed regularly with live Team Foundation Server data in order to keep requirements and development work in sync. The user must associate tasks with requirements manually, too, in order to report on requirements progress from the Team System reporting warehouse, as shown in Appendix C.

An alert can be set up on a requirements work-item change. The alert reminds the receiver to view requirements-progress and work-breakdown sheets, to stay on top of changing customer need. Work that the team is performing also is undergoing constant change. Team progress must be monitored and refreshed regularly in the requirements work-breakdown reports and views, to determine requirements impact.

As developers or testers check-in code, check-ins should be associated to related tasks, and appropriate tasks should be closed. A check-in policy is available with both Visual Studio Team System 2005 and Visual Studio Team System 2008. When the policy is set, it requires each check-in to have a related work item; otherwise, the check-in will fail. This policy helps developers and testers report progress in real time, as well as ensure that

the appropriate source-code artifacts are linked to tasks and requirements. From there, project plans and spreadsheets can be updated easily.

With the integrated tools and work-item/artifact relationships now linked, reports can be generated to monitor progress against business goals continuously. Everyone on the team now has an understanding of work and related requirements. Appendix B shows a requirements-quality report, and Appendix C shows a requirements-progress report. These reports will illustrate progress for all work that is linked to a requirement. It will not tell you whether the work is a dependency or part of the work breakdown. With Visual Studio Team System 2005 and Visual Studio Team System 2008, it is difficult to filter out different relationships. This problem will be addressed in Visual Studio Team System 2010.

### Requirements Verification

Business analysts validate end-user requirements by picking up a build of the solution and running manual acceptance tests. However, the following must be kept in mind:

- **Requirements change while the test team is testing a previous set of requirements.** The test team is unaware that the requirements have changed and that their tests no longer are providing the necessary requirements coverage. The business analyst does not have the up-to-date manual tests to provide acceptance testing that is based on the changed requirements.
- **When are the requirements ready to test?** It is difficult to know when the requirements are complete and which build to pick up.

Activity	Purpose/Timeframe	Appropriate tools	Partner tools
<b>Prototype.</b>			
<b>Update test cases.</b>	Ensure that test cases provide coverage over the most up-to-date requirements.	Test-automation tools Spreadsheets Reports on test cases/requirements	
<b>Run acceptance tests.</b>	Validate the end-user functionality that is required.	Spreadsheets Manual-testing tools	
<b>Use end-user feedback.</b>	Solicit feedback from customers on the proposed solution.	Voting tools Comments on the solutions in the tools Reports on bugs, feedback	Personify Design TeamLook

---

The traceability and alerting that are available in Visual Studio Team System 2005 and Visual Studio Team System 2008 help the team collaborate as work progresses. When the team updates tasks and tasks are linked to requirements, a query or report on requirements can show the progress of those tasks. The query or report can show which of the requirements is complete.

Team Foundation Build, which is part of Team Foundation Server, provides build reports that contain build numbers, their work-item changesets, and associated work items. From this type of report, it is easy to see which tasks, bugs or requirements are checked-in to which build. Testers and business analysts now can pick up appropriate builds, understanding which requirements, tasks, and bugs are addressed. Alerts can be set on builds, and testers can get started right away as the build is available.

With this process in place, requirements can be tested close to the time of check-in, and immediate course correction can be taken. Testers can be more productive and run the correct tests; they no longer are testing out-of-date requirements or incorrect builds. All throughout, requirements progress and quality reports are generated and communicated to the end user. With all of this up-to-date data, the business analyst can spend less time trying to understand what the development team is doing and focus, instead, on communicating with the customer, validating and updating requirements, and collaborating on requirements change.

---

## **REQUIREMENTS FUTURES WITH TEAM SYSTEM 2010**

### **Requirements Refinement**

Visual Studio Team System 2010 will intend to address the requirements-refinement process. For example, it will improve the work breakdown process by providing a way to break down child requirements or tasks from within the Team Foundation Server client. It will be easy to create hierarchies of work items and indicate interdependencies easily. The process template will define the types of relationship links that you would like to have available for your methodology (for example parent/child, predecessor/successor, dependency) and specify rules, such as required links or fields that roll-up across work items. This enables a lot of power in managing work relationships and describing how work is related:

As different work relationships are created, queries can be run on the types of links—for example:

- What requirements have no corresponding work?
- What requirement does this task meet?
- What tasks exist that don't stem from a requirement?

The same information can be generated from a report in the Team System reporting warehouse. The warehouse also can run trends over time, answering questions such as:

- How much has this requirement scope changed over the last month?
- How fast are we making progress on this requirement?

Full editing of requirement (and other work-item hierarchies) will be available in Office Excel and Office Project. Any parent/child links that are stored in Team Foundation Server will appear in the appropriate place in the tree hierarchy in Office Excel or Office Project. Modifications that are made in those Microsoft Office tools will be persisted back in Team Foundation Server when the data is published. In Office Project, you can also edit and refresh other link relationships—such as predecessor/successor—and the same data will be reflected in both Team Foundation Server and Microsoft Office. When work relationships change, the whole system is kept up to date in an integrated fashion, and team members are immediately aware of their work context and dependencies.

Team Foundation Server 2010 also will enable the management of resources that are shared across projects by integrating with Office Project Server. Office Project Server provides an enterprise resource pool, in which calendar, vacation, and capacity information is collected on each individual. As the people are allocated to multiple projects, their full work utilization across all of the projects is understood. People now can be load-balanced with team members, and their contributions to all projects taken into consideration.

In addition, we intend to create a system that makes it easy to author and run manual tests, make it very easy to link tests to requirements, and generate a running set of reports that indicate requirements quality, including manual and automated tests.

---

All of these improvements will make refining requirements a smooth process—one in which the right work relationships are set up naturally without extensive user intervention. People can get their work done in the tools in which they are productive. Managers and business analysts can view work progress and quality progress on requirements easily. Also, they can use traceability throughout the system to understand both the context of the work and any changes.

### **Requirements Implementation**

With Visual Studio Team System 2005 and Visual Studio Team System 2008, team members understand that work is related, but they might not understand how. Visual Studio Team System 2010 will provide customizable link types that will make it much easier to create meaningful link relationships between requirements and other work in the system. Team members will better understand work dependencies and how work stems from a requirement, without having to go to a specific spreadsheet or project plan. For example, we can see now the tasks that make up a requirement and other requirements that are dependent. We expect to provide reports and views that illustrate how work is related to each other, to help people better understand the context of their work and the effect of a change.

We expect also to provide rules around links, to enable work rollup or dependency acceptance. These rules will make sure that the right link relationships are set up and keep the maintenance of the links to a minimum. The rules should be customizable and extensible, so that processes can be enacted in the tools more easily.

In addition, we expect to improve the subscriptions and notification system in Visual Studio Team System 2010 in order for users to subscribe more easily to alerts that are meaningful to them, and to eliminate the noise of receiving too many “work item changed” alerts.

As a result, Visual Studio Team System 2010 should make it easy for the team to follow the process and maintain work relationships easily throughout the application life cycle. Both business analysts and end users will be more connected to the work of the development team with reports that are up to date and reflect the work of the team accurately. Team members are immediately aware of requirements change and can assess change impact more easily.

### **Requirements Validation**

Visual Studio Team System 2010 will make it easy to track and execute manual tests, and to link these tests to requirements. The business analyst now will have tools to run manual acceptance tests against builds and track the results. The results will be available in the Team System reporting warehouse—enabling full requirements-quality reports over all test cases.

---

## REQUIREMENT PARTNERS

### Borland CaliberRM

Borland® CaliberRM™ is an enterprise-software requirements-management tool that facilitates collaboration, impact analysis, and communication—enabling software teams to deliver on key project milestones with greater accuracy and predictability.

CaliberRM integrates seamlessly into Team System and provides a rich requirements-management environment that facilitates requirements management with a hierarchical, tree-view storage and display of requirements.

Team System, coupled with CaliberRM, helps teams deliver higher-quality products in less time by making that sure requirements updates are provided to the correct team members, and that the linked code, tests, and development assets are updated to reflect these changes.

[Company Web Site](#) | [CaliberRM Overview](#) | [Demo/Walkthrough](#) | [White Paper on Using It with Visual Studio Team System](#)

### Compuware Optimal Trace

Compuware Optimal Trace™ captures software requirements from the perspective of the user—complete with visual storyboards and traceable relationships—throughout the project life cycle to business needs.

Optimal Trace Enterprise Edition integrates with Team System, which enables Team System users to leverage the advantages of Optimal Trace—resulting in accurate and complete business and system requirements. This enables the tracing back of all design and QA activities in Team System to requirements, and full end-to-end traceability right through the project life cycle from conception to deployment.

The Optimal Trace Integration enables the automatic population of Team System “work items” within Team Foundation Server, which avoids the need for manual input and ensuring that complete and accurate requirements form the basis for the project activities. The integration also provides the ability to update into the Team Foundation Server project any requirements changes that are made in Optimal Trace. These updates preserve the integrity of any other data that is associated with the work item, including such aspects as estimated effort, which means that change management is facilitated fully during the life cycle.

[Company Web Site](#) | [OptimalTrace Overview](#) | [Demo/Walkthrough](#) | [Webcast on Using It with Visual Studio Team System](#)

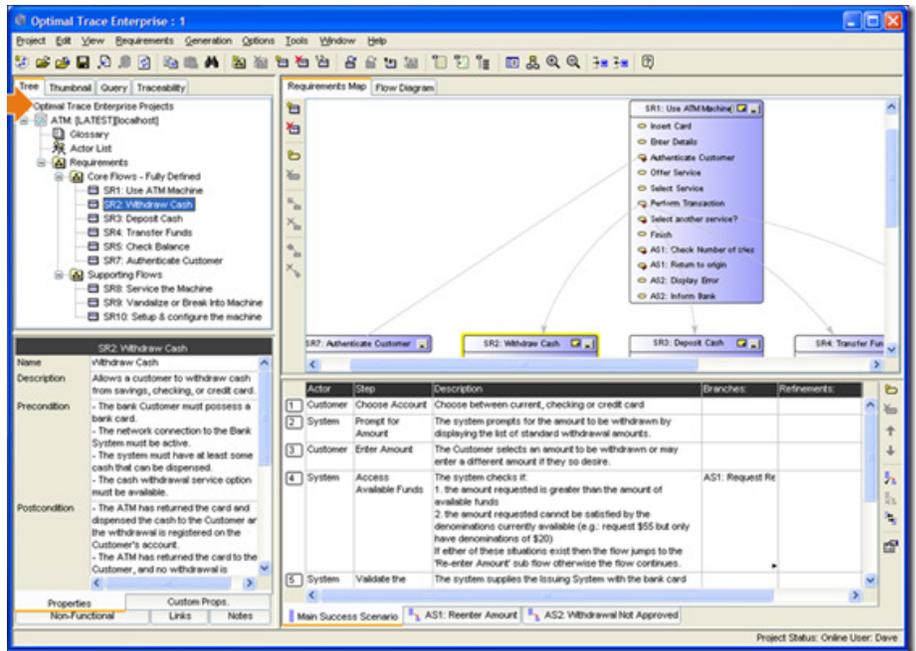


Figure 5. Example Compuware Optimal Trace Enterprise Edition screen

## Mindjet Requirements Manager

Mindjet Requirements Manager for Team System provides developers with a quick, simple method to gather requirements by using the mind-mapping paradigm; then, they can create work items in Team System and track the project progress in a way that is optimized for requirements elicitation and quick ad-hoc collection of requirement in meetings. Based on Mindjet MindManager's visual interface, Mindjet Requirements Manager enables teams to capture, organize, and manage requirements; the bidirectional link to Team System simplifies work-item tracking.

[Company Web Site](#) | [Requirements Manager Overview](#) | [Demo/Walkthrough](#)

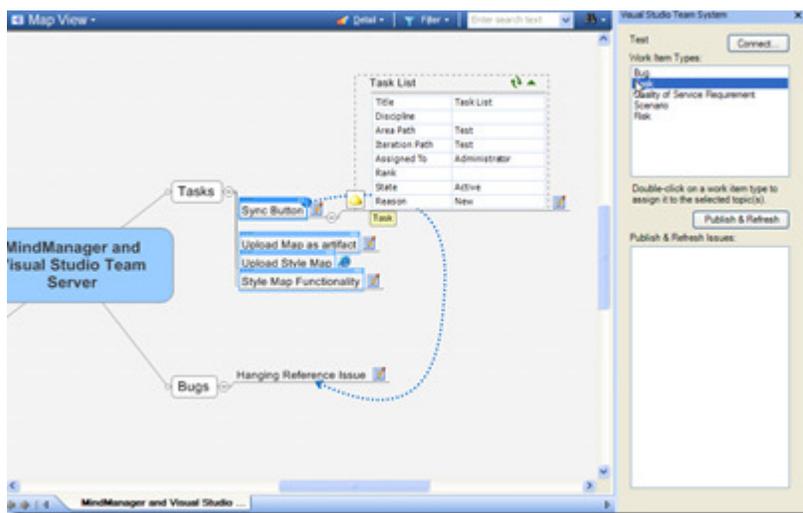


Figure 6. Example Mindjet MindManager visual interface

## Personify Design TeamSpec and TeamLook

Personify Design TeamSpec™ provides a way to connect requirements specifications in Office Word to Team Foundation Server. TeamSpec enables users to create structured Office Word documents out of Team Foundation Server project artifacts (change requests, bugs, scenarios, and tasks, among other work-item types) or push content from existing Office Word documents into Team Foundation Server.

Personify also provides TeamLook™, which is a convenient software-project dashboard that enables all stakeholders in a software-development project to communicate and collaborate on Visual Studio Team System 2005 projects from within Office Outlook.

[Company Web Site](#) | [TeamSpec Overview](#) | [Demo/Walkthrough](#)

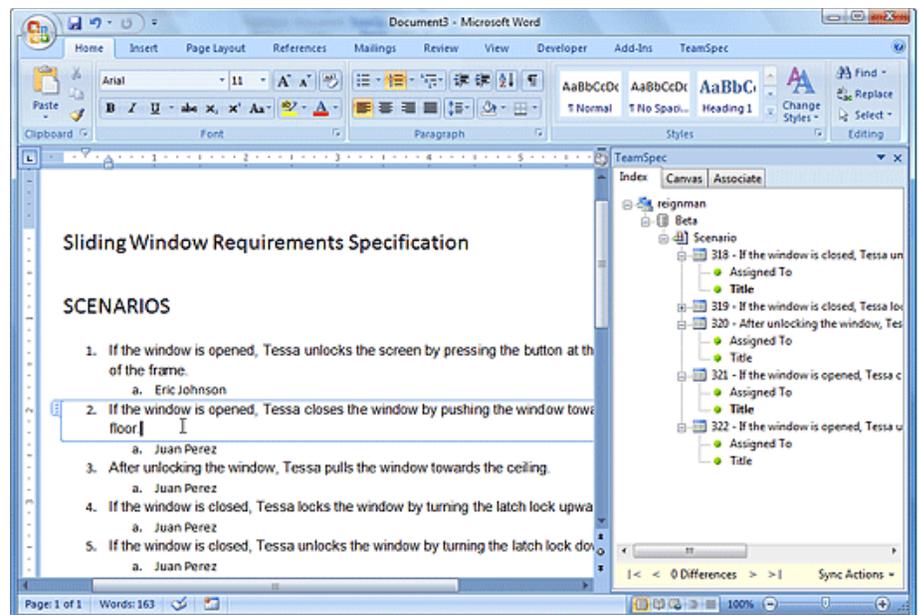


Figure 7. Example Personify Design TeamSpec and Office Word screen

## Ravenflow RAVEN

Ravenflow provides a solution—called RAVEN™—that assists business analysts with requirements elicitation, analysis, specification, and validation. The RAVEN syntactic-analysis software parses plain business-English text, and automatically generates activity diagrams and responsibility diagrams—providing immediate graphical feedback for detecting requirements errors and understanding application flow.

Ravenflow's Team System integration enables requirements that are captured in Ravenflow to be published as work items in Team Foundation Server.

[Company Web Site](#) | [RAVEN Overview](#) | [Demo/Walkthrough](#)

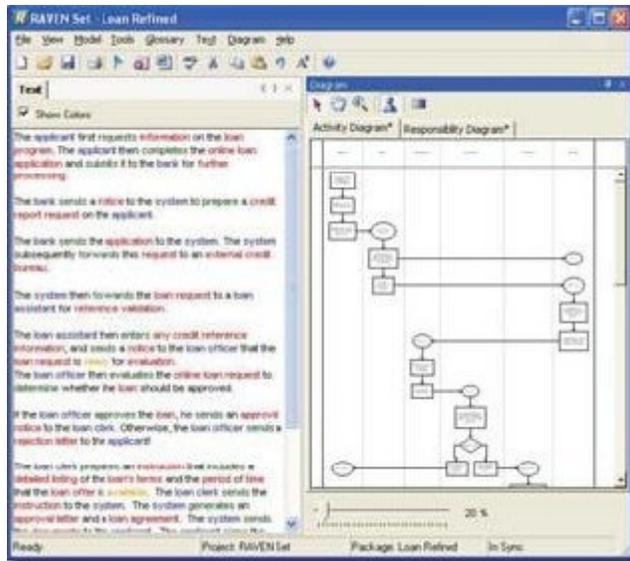


Figure 8. Example RAVEN screen, with diagram

### stpsoft stpBA and Quew

stpsoft stpBA Storyboarding for Visual Studio Team System 2005 enables a business analyst or analyst developer to capture, elaborate, and validate requirements and scenarios in a team project through GUI storyboarding by using Microsoft Visio-compatible graphics. The tool integrates seamlessly with Team System process templates, and generates screen-flow diagrams, HTML storyboards, UI specifications, functional specifications, Team System work items, and test scripts. stpsoft also provides a requirements tool—called [stpsoft Quew](#)—which is a lightweight documents-based requirements tool.

[Company Web Site](#) | [stpBA for Visual Studio Team System Overview](#) | [Demo/Walkthrough](#)

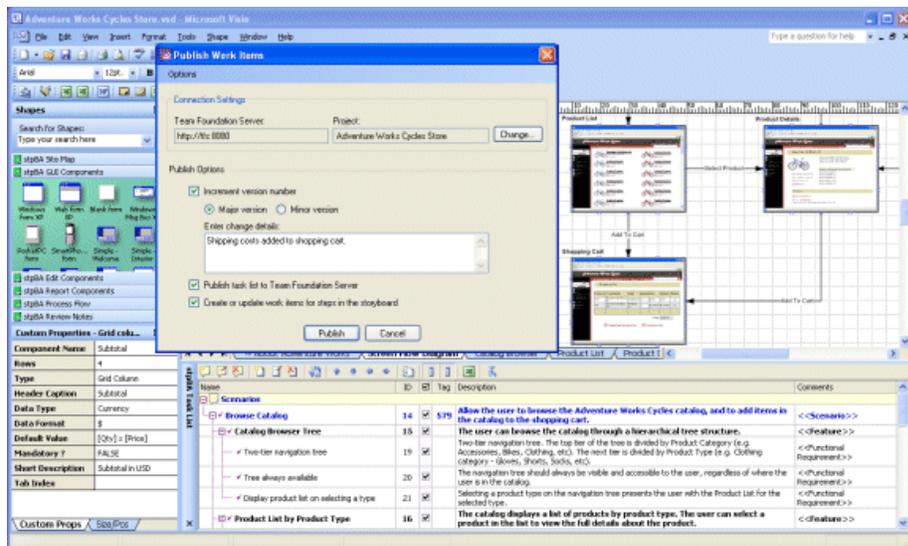


Figure 9. Example Publish Work Items dialog box in stpBA

---

## **Techno Solutions Visual Use Case and TopTeam Analyst**

Techno Solutions markets a suite of ALM tools that includes two tools that can be used for requirements development. Each of these tools integrates with Team Foundation Server.

Visual Use Case™ is a tool for authoring use-case flow-of-events and drawing use-case diagrams. TopTeam Analyst™ is a requirements-management tool that provides tools to document, prioritize, and manage requirements. It uses a hierarchical requirements structure, and provides rich-text–editing capability and traceability diagramming tools and editors.

[Company Web Site](#) | [Visual Use Case Overview](#) | [TopTeam Analyst Overview](#)

## **Holocentric Modeler**

From the Holocentric Modeler 5.2 brochure: “Use Holocentric Modeler 5.2 to identify business-process inefficiencies and design better practices. Realistic scenarios can be captured which include all aspects of the process, such as process step durations and lag times, frequencies and volumes, salaries and skill levels. Presentation is important. Use alternate diagram notations to suit your audience such as swim lanes, free-form representation and more detailed diagram views showing duration and lag times. Processes may be analyzed and changes simulated to enable the optimum results to be achieved. Visualize paths under analysis in your model by highlighting scenarios.”

[Company Web site](#) | [Holocentric Modeler 5.2 Brochure](#) | [Demonstrations](#)

## APPENDIX A: WORK BREAKDOWN IN OFFICE EXCEL

The indents in the Title column are preserved only in the spreadsheet. In other words, the inferred parent-child relationship of work items is not saved on Team Foundation Server. Do not discard the .xls file. To view the effect of work progress on requirements, this spreadsheet will need to be refreshed regularly.

To link requirements to tasks, pull up the **Links and Attachments** dialog box and associate requirements to tasks manually. That will enable the same information to be reported in the Visual Studio Team System reporting warehouse.

Project: Rosario Server: http://vstfdevdivir.8080 Query: All Deliverables List type: Flat			
ID	Title	Column1	Assigned to
1357	the system must accept at least one character per second	Requirement	Lori Lamkin
1358	input screen	Task	Lori Lamkin
1359	fix existing performance issues	Task	Lori Lamkin
1694	the system must have a Web interface	Requirement	Lori Lamkin
2145	logon screen	Task	Lori Lamkin
2395	welcome screen	Task	Lori Lamkin
2444	order screen	Task	Lori Lamkin
2456	inventory browsing	Task	Lori Lamkin
2556	users should be able to place orders	Requirement	Lori Lamkin
2767	back-end database	Task	Lori Lamkin
2926	update inventory	Task	Lori Lamkin

Figure 10. Example work breakdown in Office Excel

## APPENDIX B: REQUIREMENTS-QUALITY REPORT

Test results can be associated with requirements by either linking them together in the work item form or associating a test-run result with a requirement. After this association has been created, pull the Test Results perspective from the Visual Studio Team System cube. You can do this either while creating an .rdl file or by binding the data source into Office Excel.

Figure 11 shows the cube bound to Office Excel, and a report on requirements and the test-run results.

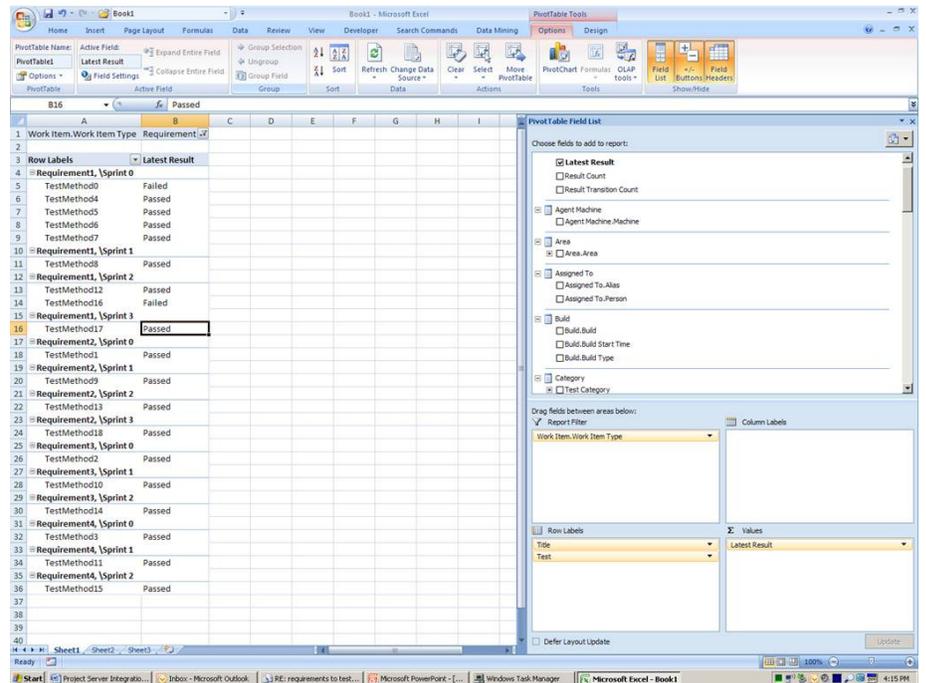


Figure 11. Example requirements-quality report, with cube bound to Office Excel

## APPENDIX C: REQUIREMENTS- PROGRESS REPORT

Figure 12 shows the linkage between requirements and tasks by using some sample data to show how many tasks remain for each requirement. The users had to ensure that the requirement and task work items were linked together in Team Foundation Server work-item forms, before such a report is possible. This data is pulled from the “Related Work Items” cube, which has a primary attribute and a related attribute.

In the example in Figure 12, you can see that “Work Item Type” is filtered on “Requirement” and “Related Work Item Type” is filtered on “Task”—eliminating from this view any other possible work-item links.

Work Item Type	Work Item Type	Requirement			
Related Work Item Type	Work Item Type	Task			
Related Project	Project	Formal_620			
Project	Project	Formal_620			
Related Count					
Title	Identifier	Identifier	Title	State	
Requirement1, Iter 1-0	2	10	Task 1, Scenario=Scenario_0_1,iteration=Iter 1-0	Active	
		17	Task 8, Scenario=Scenario_0_1,iteration=Iter 1-0	Active	
		20	Task 11, Scenario=Scenario_0_1,iteration=Iter 1-0	Active	
		24	Task 15, Scenario=Scenario_0_1,iteration=Iter 1-0	Active	
		27	Task 18, Scenario=Scenario_0_1,iteration=Iter 1-0	Active	
		43	Task 34, Scenario=Scenario_0_1,iteration=Iter 1-0	Active	
Requirement2, Iter 1-0	3	13	Task 4, Scenario=Scenario_0_2,iteration=Iter 1-0	Active	
		18	Task 9, Scenario=Scenario_0_2,iteration=Iter 1-0	Active	
		22	Task 13, Scenario=Scenario_0_2,iteration=Iter 1-0	Active	
		23	Task 14, Scenario=Scenario_0_2,iteration=Iter 1-0	Active	
		32	Task 23, Scenario=Scenario_0_2,iteration=Iter 1-0	Active	
Requirement3, Iter 1-0	4	28	Task 19, Scenario=Scenario_0_3,iteration=Iter 1-0	Active	
		37	Task 28, Scenario=Scenario_0_3,iteration=Iter 1-0	Active	
		44	Task 35, Scenario=Scenario_0_3,iteration=Iter 1-0	Active	
Requirement4, Iter 1-0	5	25	Task 16, Scenario=Scenario_0_4,iteration=Iter 1-0	Active	
		42	Task 33, Scenario=Scenario_0_4,iteration=Iter 1-0	Active	
Requirement5, Iter 1-0	6	12	Task 3, Scenario=Scenario_0_5,iteration=Iter 1-0	Active	
		14	Task 5, Scenario=Scenario_0_5,iteration=Iter 1-0	Active	
		15	Task 6, Scenario=Scenario_0_5,iteration=Iter 1-0	Active	
		16	Task 7, Scenario=Scenario_0_5,iteration=Iter 1-0	Active	
		21	Task 12, Scenario=Scenario_0_5,iteration=Iter 1-0	Active	
		31	Task 22, Scenario=Scenario_0_5,iteration=Iter 1-0	Active	
		34	Task 25, Scenario=Scenario_0_5,iteration=Iter 1-0	Active	
36	Task 27, Scenario=Scenario_0_5,iteration=Iter 1-0	Active			
40	Task 31, Scenario=Scenario_0_5,iteration=Iter 1-0	Active			
Requirement6, Iter 1-0	7	11	Task 2, Scenario=Scenario_0_6,iteration=Iter 1-0	Active	
		19	Task 10, Scenario=Scenario_0_6,iteration=Iter 1-0	Active	
		35	Task 26, Scenario=Scenario_0_6,iteration=Iter 1-0	Active	
		39	Task 30, Scenario=Scenario_0_6,iteration=Iter 1-0	Active	
Requirement7, Iter 1-0	8	26	Task 17, Scenario=Scenario_0_7,iteration=Iter 1-0	Active	
		30	Task 21, Scenario=Scenario_0_7,iteration=Iter 1-0	Active	
		33	Task 24, Scenario=Scenario_0_7,iteration=Iter 1-0	Active	
		38	Task 29, Scenario=Scenario_0_7,iteration=Iter 1-0	Active	
		41	Task 32, Scenario=Scenario_0_7,iteration=Iter 1-0	Active	
Requirement8, Iter 1-0	9	29	Task 20, Scenario=Scenario_0_8,iteration=Iter 1-0	Active	
Grand Total					

Figure 12. Example requirements-progress report