

Team Foundation Server Planning Guide

Visual Studio ALM Rangers



Microsoft



Visual Studio

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 document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Microsoft grants you a license to this document under the terms of the Creative Commons Attribution 3.0 License. All other rights are reserved.

© 2013 Microsoft Corporation.

Microsoft, Active Directory, Excel, Internet Explorer, SQL Server, Visual Studio, and Windows are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Table of Contents

Foreword	5
Introduction	6
What's changed?.....	8
TFS 2012.....	8
TFS 2013.....	8
Defining your TFS Strategy	9
Understanding TFS Topologies.....	10
Deciding on service versus on-premises server infrastructure	11
Determining on-premises server infrastructure capacity planning strategy	13
Considering Upgrade and Migration.....	17
Considering Virtualization	19
High Availability Strategies	19
Build Server Considerations.....	20
Proxy Server Considerations.....	21
Defining your Team Project Collection Strategy	23
Understanding Team Project Collections	24
Key Decisions – Single versus Multiple Team Project Collection Strategy	26
Deciding on Team Project Collection Strategy.....	26
Defining your Team Project Strategy	29
Understanding Team Projects	30
Team Project Boundaries.....	31
Team Project Constraints.....	32
Key Decisions – Single versus Multiple Team Project Strategy	33
Deciding on a Team Project Strategy	34
Defining your Team Strategy.....	38
Understanding Teams	39
Team Boundaries.....	39
Team Constraints	39
Key Decisions – Single versus Multiple Team Strategy	40
Area Paths and Teams.....	40
Deciding on Team Strategy	40
Defining your Disaster Recovery (DR) Strategy	44
Understand DR Strategies	44
Understand DR Avoidance.....	46
DR Recovery Walkthroughs	51
Large and Complex Project considerations	53

Planning Guide – Foreword

Understanding Large Projects	54
Deciding on a Large Project Strategy	55
Strategy 1 - Single Team Project Collection examples	57
Strategy 2 - Multiple Team Project Collections examples	58
Questions & Answers	62
Real World Reference Stories	64
Scenario 1 – Size doesn't matter	65
Scenario 2 – Getting complex	66
Scenario 3 – New releases every day	67
Scenario 4 – Across the globe	68
Scenario 5 – No time for refactoring	69
Scenario 6 – Disaster by hotfix - Vladimir's DR Experience	70
Scenario 7 – Aborted Warehouse Processing - Thorsten's DR Experience	71
Appendix	73
Enabling and using TFS Logging Examples	73
Working with TFS Performance Counters	77
DR Recovery Walkthroughs	80
Authoring Reports	97
Sample Maintenance Scripts	100
Quick Reference Sheets	103

Foreword

In 2005, within six months of joining the Team Foundation Server (TFS) v1 team, I was given the task to go to Ferrari and install TFS. The assignment was actually more than a simple installation, involving server capacity planning and developer training all crammed within 4 days. As I soon found out, designing and implementing a deployment that can scale as an enterprise organization either increases in size, increases its project complexity or adds more departments is a challenging task. There are a lot of variables, opinions and at times “plain guesses” - I mean estimations - at play. During that trip I still vividly remember working with one of the Ferrari managers until 1AM creating a spreadsheet with data that was going to assist us in making the right deployment decisions. We collected data about # users, # of projects, the size of the version control tree and the expected load of their build system. It was fun!

The product has evolved and matured since those early days. We added a scale-out solution, optimized for very large source trees, introduced teams and authored a number of ALM training resources. Clearly we have grown, but so has the need and complexity of capacity and project planning. An updated, thorough and technical document is needed. Enter the Rangers and the creation of a comprehensive solution that marries the product team’s knowledge with all of the field’s experience. The document will provide you with the concepts, tools and best practices needed to tackle the heterogeneous nature of your customer engagements. Just like in 2005, it continues to be fun seeing us grow!

Mario Rodriguez (*Senior Program Manager, TFS Infrastructure*)

Introduction

This guidance delivers practical and scenario based guidance for the implementation of Team Foundation Server (TFS). We guide you through the decisions whether to have one or more Team Foundation Servers, one or more Team Project Collections, one or more Team Projects and one or more Teams, based on scenarios and implications of each decision. We conclude with disaster recovery planning, frequently asked questions and a collection of real world reference stories.

NOTE

This version (v1.3) of the guidance is focused on Team Foundation Server **2012** and **2013**, whereby most of the concepts and guidance are backwards compatible—often identical—with Team Foundation Server **2010**.

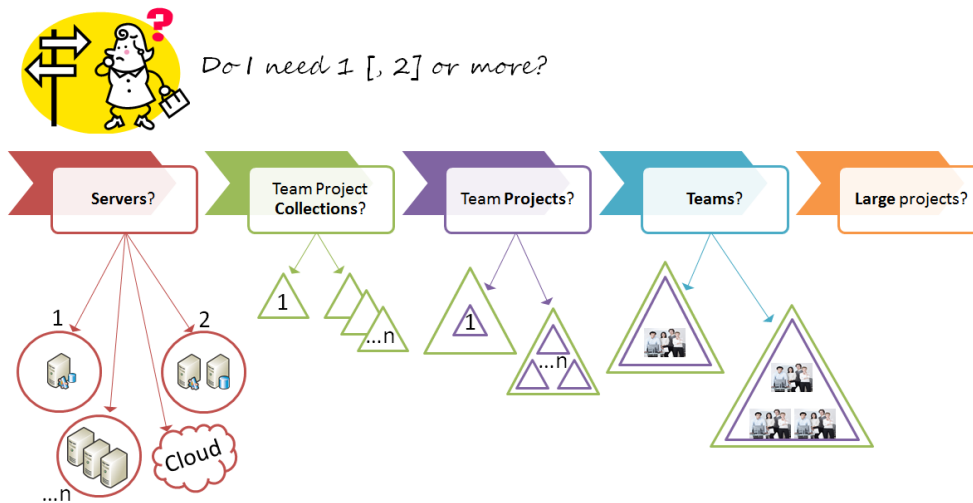


Figure 1 - Making decisions whether we need one, two or more or "stuff"

Visual Studio ALM Rangers

The Visual Studio ALM Rangers are a special group with members from the Visual Studio Product group, Microsoft Services, Microsoft Most Valuable Professionals (MVP) and Visual Studio Community Leads. Their mission is to provide out-of-band solutions to missing features and guidance. A growing Rangers Index is available online¹.

Contributors

Chris Wishart, Daniel Meixner, Ed Holloway, Francisco Fagas, Gregg Boer, Guy Teverovsky, Jeff Levinson, Jim Szubryt, Joakim Karlsson, John Berman, Lennart Jansson, Mario Rodriguez, Mike Fourie, Pramod Vasanth, Prasanna Ramkumar, Stefan Mieth, Thorsten Dralle, Tiago Pascoal, Tim Star, Tina Erwee, Tommy Sundling, Willy-Peter Schaub

¹ <http://aka.ms/vsarindex>

How to Use This Guidance

Each section of this guidance has a common content structure:

- **Topic Overview** ... presents a brief overview of the intent of the relevant section and mentions the personas that are relevant (interested) in the section.
- **Deciding what is Important in this section** ... a quick reference table that give you are quick map into the topics within the section, based on your interests.
- **Understanding the topic** ... delves into the topic and covers topics such as constraints.
- **Scenarios** ... presents some common scenarios / strategies that can be used as reference.



Use the TFS Planning Guidance Hands-on Lab as a practical walkthrough and companion when perusing this guidance and the Quick Reference Posters and Cheat Sheets for quick reference.

Personas

Refer to [Visual Studio ALM Rangers Personas and Scenarios](#)² for more information about the personas and customers profiles referenced in this guide.

Additional ALM Rangers Resources

Understanding the ALM Rangers – <http://aka.ms/vsarunderstand>

Visual Studio ALM Ranger Solutions – <http://aka.ms/vsarsolutions>

² <http://go.microsoft.com/fwlink/?LinkID=230942>

What's changed?

TFS 2012

This section introduces you to the impact that TFS 2012 has on the guidance around planning TFS environments.

Capacity Planning

With the exception of support for Windows Azure, there are no capacity planning changes compared to TFS 2010.

Refer to [Team Foundation Service Practical Guidance](#)³ for more details and guidance around the Team Foundation Service scenario.

See **Defining your TFS Strategy** on page 9 for details.

Team Project Collections

No changes with Team Project Collection planning compared to TFS 2010.

See **Defining your Team Project Collection Strategy** on page 23 for details.

Team Projects

No changes with Team Project planning compared to TFS 2010.

See **Defining your Team Project Strategy** on page 29 for details.

Teams

The concept of Teams is new and introduces new planning guidance.

See **Defining your Team Strategy** on page 38 for details.

TFS 2013

TFS 2013 has not introduced any changes in terms of the server, team project collection, team project, teams, or disaster recovery planning. This guide has been reviewed for compliance and minor updates have been added for SharePoint 2013. Otherwise the concepts and guidance covered herein are backwards compatible—often identical—with Team Foundation Server **2010** and **2012**.

³ <http://go.microsoft.com/fwlink/?LinkID=230945>

Defining your TFS Strategy

This section introduces you to the common TFS topologies and the planning steps, which depend largely on the current and forecasted size of your teams.

If you are new to TFS we recommend that you first understand the TFS **Architecture** and read the TFS **Installation** and **Administration** guides, as well as other publications mentioned under references on page 22.

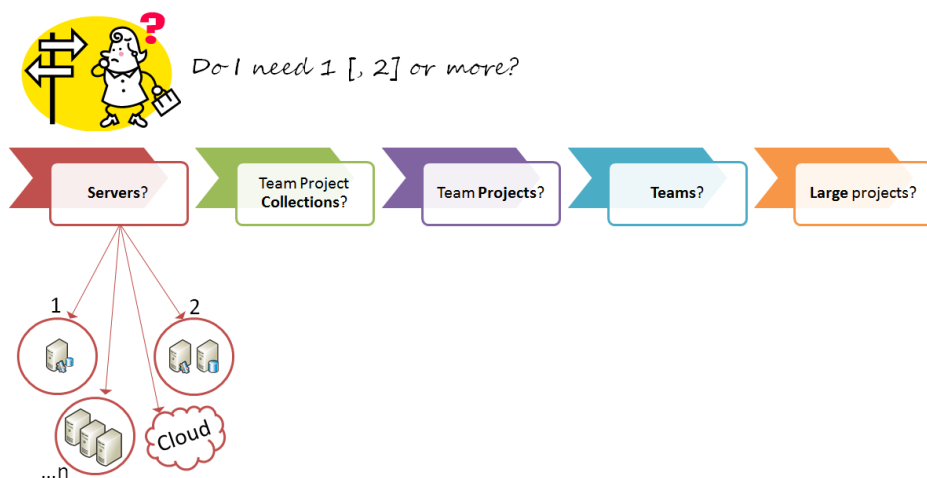
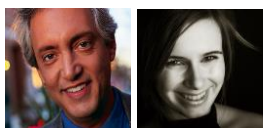


Figure 2 - TFS topology planning

Refer to Hands-on Lab, section “**Stepping through the planning of a Server Strategy**”, for a walkthrough of this section.

This section is focused on **Dave** the TFS administrator and **Jane** the infrastructure specialist.



I would like to ...	Page(s)
• Understand the server topologies	10
• Decide whether to use a hosted or on-premises infrastructure	11
• Understand the deployment options	13
• Plan the deployment based on number of users	13
• Plan the deployment of Application tier (AT) servers when scaling out	14
• Plan the deployment based on number of Team Project Collections	17
• Plan the deployment based on number of Team Projects	17
• Consider upgrading or migrating your infrastructure	17
• Consider virtualizing your infrastructure	19

Table 1 - Deciding what information is important with TFS planning

Understanding TFS Topologies

There are a number of TFS deployment options and we will begin by presenting the most common approaches and then exploring key decision points which will allow you to make a better decision.

There is the hosted service, the single server and the multi-server deployment option, whereby the latter includes both scaling up and out.

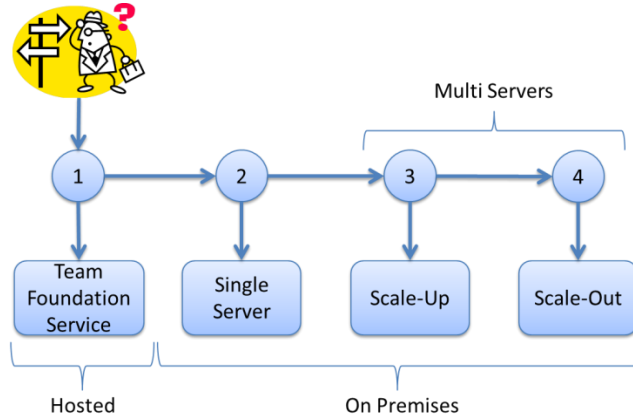


Figure 3 - Deciding which infrastructure suits your needs best

Single Server



The single-server deployment option places the application and the Data tier on a single server, whereby build and proxy servers are optional features and can be deployed on the same or separate servers.

Multi-Server (Scale Up)



The multi-server (scale up) deployment option usually starts with the application and the Data tier on separate servers, whereby the build and proxy servers are again optional, but typically deployed on separate servers in this scenario. When you are “scaling up,” you are adding resources to a single node in the system, typically adding more CPU, Memory or Disk Space.

Multi-Server (Scale Out)



In the “scale out” model, a different approach is taken. When you add resources, you add a new node to the system to distribute load and achieve greater capacity. An example would be adding a new computer to the Application tier in order to distribute user request load. As computer prices drop and performance continues to increase, low cost “commodity” systems can be easily leveraged in a grid/cluster to achieve large amounts of computer power and performance. The multi-server (scale out) deployment option adds additional redundancy and performance by adding additional Application tier and Data tier servers. The Data tier is typically implemented using SQL Server Instances and the Application tier by network load balancing (NLB) two or more Application tier servers.

Team Foundation Service



The Team Foundation Service deployment option uses the Microsoft cloud platform hosted in Microsoft data centers or in-place using the Windows Azure Platform Appliance.

You can refer to the Visual Studio ALM Rangers [Visual Studio Team Foundation Service Guidance](#) for more information and guidance on Team Foundation Service.

TFS on Azure IAAS

A special case for the on-premises scenario is to host your environment in your private cloud, as outlined in the companion “TFS on Azure IAAS” guide. As shown below, this scenario combines the best of both the hosted and the on-premises environment.

NOTE

[ALM Rangers dogfooding journal of the Team Foundation Service](#)⁴ documents the use of the Team Foundation Service by the ALM Rangers as part of their 24x7x365 dog-fooding exercise.

Deciding on service versus on-premises server infrastructure

The first decision we need to make is whether or not the Team Foundation Service is a viable deployment option.

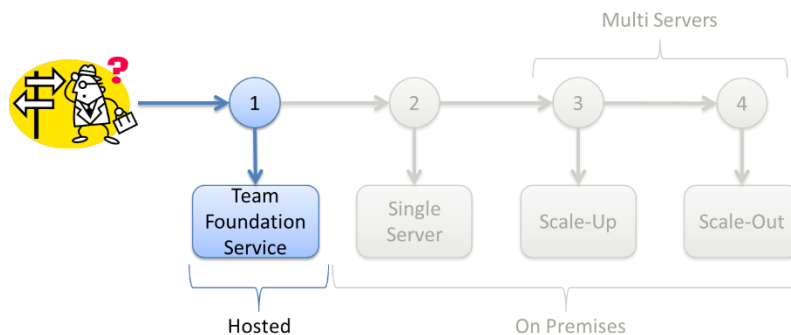


Figure 4 - Exploring the Team Foundation Service deployment option

Key Decisions – Hosted Service versus On-Premises Server

😊 Feasible 😞 Limitations ☹ Not feasible

		Service	Server	TFS on Azure IAAS
TFS	Process Template Customization	☹ Not generally available at the time of writing this guidance.	😊	😊
	Work Items, Source Control and Build features	😊	😊	😊
	Agile Projects Management	😊	😊	😊
	Test Case Management	😊	😊	😊
	Heterogeneous Development	😊	😊	😊
	Near-zero setup and administration	😊	☹ Administration of the operational infrastructure is a huge cost when using on-premises servers, even if initiatives such as the VM Factory can introduce near-zero setup.	😊 Includes administration of the operational infrastructure which incurs cost. Also includes setting up of Azure infrastructure and maintenance of the infrastructure.

⁴ <http://aka.ms/vsarcloudhome>

Planning Guide – Defining your TFS Strategy







































		Service	Server	TFS on Azure IAAS
	Collaborate from anywhere		 On-premises infrastructure may introduce premise-specific authentication and accessibility constraints.	 Azure infrastructure may introduce Azure-specific authentication and accessibility constraints.
	Virtual Test Lab Management	 Not available at the time of writing this guidance.		
	SharePoint Integration	 Not available at the time of writing this guidance.		
	Data Warehouse & Reporting	 Not available at the time of writing this guidance.		
Environmental	No infrastructure needed to host TFS			 TFS on Azure may require additional time to setup the infrastructure.
	On-premises infrastructure with unpredictable usage spikes		 On-premises infrastructure must be engineered for extra scalability and fault tolerance to accommodate unpredictable spikes, which is a costly investment that may not meet ROI criteria.	 TFS on Azure infrastructure must be engineered for extra scalability and fault tolerance to accommodate unpredictable spikes, which is a costly investment that may not meet ROI criteria.
	Anticipated usage that exceeds on-premises infrastructure capacity		 On-premises infrastructure must be engineered to allow scalability without downtime, which is a costly investment that may not meet ROI criteria.	 TFS on Azure infrastructure must be engineered to allow scalability without downtime, which is a costly investment that may not meet ROI criteria.
	Need for total control of infrastructure for auditing			
	Multiple partner companies needing an integrated solution		 On-premises infrastructure may introduce premise-specific authentication and accessibility constraints.	 TFS on Azure infrastructure may introduce premise-specific authentication and accessibility constraints.
	Data in motion / at rest constraints for compliance purposes (HIPPA, PCI, etc.), such as the strictly regulated gaming industry			
Customer Types	Humongous Insurance			
	Trey Research			
	Consolidated Messenger			

Table 2 - Team Foundation Service key decisions and viability

Determining on-premises server infrastructure capacity planning strategy

To decide which of the in-place deployment options are viable, you need to consider the advantages of each option and the number of users and projects you will need to support, which typically varies between the **Humongous Insurance**, **Trey Research** or **Consolidated Messenger** customer types.

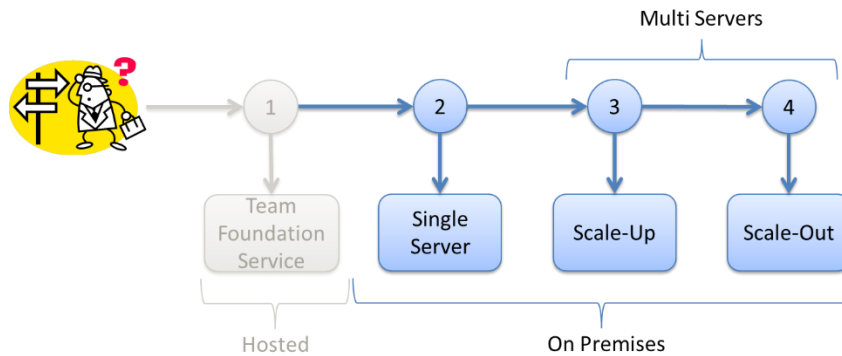


Figure 5 - Exploring the on-premises deployment option

Understanding the advantages of each deployment option

Some key decisions will impact your decision whether to pursue one or more of the deployment options.

Key Decisions	Single Server ⁵	Scale-Up	Scale-Out
I need an easy to use and shareable demo environment	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I need simplicity in terms of management	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I need high availability and fail-over support	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I need high scalability	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Table 3 – On-Premises Server deployment key decisions and viability

NOTE

We are not including the basic configuration of TFS in this guidance. It could be considered for the single server deployment scenario, if and only if the integration with SharePoint products or the reporting features are not required.

Real-World “Beef” Factor

WARNING

The hardware specifications documented herein are the recommended bare-metal **minimum** specifications!

The section **Real-World Reference Stories** on page 5 summarizes a few of the real-world scenarios we have received, which demonstrate that the minimum hardware specifications might not suffice in the real-world, especially if performance and scalability is a prime factor.

⁵ ☒ viable ☐ feasible ☒ not viable

Planning Guide – Defining your TFS Strategy

NOTE

This guidance includes a planning workbook, “**TFS Project Guidance - Capacity Planning**” which you can use to get feedback about the recommended deployment options and hardware specifications, based on current and anticipated number of users.

Visual Studio ALM Rangers - On-Premises Capacity Planning Quick Reference Poster Companion Workbook
Version 2012.02.24

NOTE: This is a simplified model for educational purposes and does not replace expert consulting. Real world requirements and infrastructure environments can be much more complex.

NUMBER OF ACTIVE TEAM FOUNDATION SERVER USERS

Maximum expected users: 0
Current users: 0

Real-World (Beef) Factor: 0%

INFRASTRUCTURE ARCHITECTURE	Single Server ATDT	Scale Up AT + DT	Scale Out AT + DT
Feasible recommendation for current?	Yes	Yes	No
Feasible recommendation for future?	Yes	Yes	No
Preferred Architecture?	Yes	Yes	No
Server Configuration Type	Low-End	Low-End	High-End
Maximum Current Active Collections (per SQL Server instance)	5	75	Scale Unit
Maximum Future Active Collections (per SQL Server instance)	5	75	
Estimated number of Application Tier (AT) servers	1	1	
Estimated requests per second (rps) in total	0	0	0
Estimated requests per second (rps) on AT server			

The **Real-World “Beef” Factor** is a percentage used to add more resources to the recommended minimum specifications when working with the companion capacity planning workbook. We recommend that you experiment with the minimum specifications in your test environment(s) to determine the appropriate Real-World (Beef) Factor for your environment.

Planning Multiple Application Tier (AT) Servers – Option 1 (recommended)

The TFS Project Guidance – Capacity Planning workbook allows you to determine the suggested number of Application tier servers (ATs), when you are considering a scale-out scenario.

Visual Studio ALM Rangers - On-Premises Capacity Planning Quick Reference Poster Companion Workbook
Version 2012.02.09

NOTE: This is a simplified model for educational purposes and does not replace expert consulting. Real world requirements and infrastructure environments can be much more complex.

NUMBER OF ACTIVE TEAM FOUNDATION SERVER USERS

Maximum expected users: 3000
Current users: 100

Real-World (Beef) Factor: 0%

INFRASTRUCTURE ARCHITECTURE	Single Server ATDT	Scale Up AT + DT	Scale Out AT + DT
Feasible recommendation for current?	Yes	Yes	Yes
Feasible recommendation for future?	No	Yes	Yes
Preferred Architecture?	No	Yes	Yes
Server Configuration Type	Low-End	High-End	Scale Unit
Maximum Current Active Collections (per SQL Server instance)	5	75	75
Maximum Future Active Collections (per SQL Server instance)	10	75	75
Estimated number of Application Tier (AT) servers	1	1	1 - 2
Estimated requests per second (rps) in total	18	23	21 - 675
Estimated requests per second (rps) on AT server			21 - 338

Figure 6 – Calculating the number of Application tier (AT) servers in a scale-out scenario

The workbook is based on the following two tables, which summarize the configuration and the request-per-second (RPS) guidelines:

Configuration	RPS	Conservative Users
Low End Configuration (Single Server)	92	250
Standard Configuration (Single Server)	180	500
Scale Unit Configuration (Dual Server)	476	2200
High End Configuration (Dual Server)	730	3600

Table 3 - Server configuration styles

Planning Guide – Defining your TFS Strategy

Team Size	RPS per User	Needed (RPS)
250	0.175	44
500	0.175	88
3500	0.225	788
4500	0.2625	1181

Table 3 – Request per second (rps) per AT server

NOTE

This guidance and the capacity planning workbook are intended to be used to calculate recommendations. It is very important that you **validate** any recommendations in **your** environment.

Planning based on number of users – Option 2

Another key decision that impacts the capacity planning and choice of deployment option are the number of active users you need to support for the TFS infrastructure for your organization. The following table summarizes the maximum recommended users for three deployment options and a variety of example specifications.

Deployment Option	REF	Example specification	Physical Hardware Max users
Single Server	1	1 x ATDT: 1 Processor, 2GB RAM, 125GB Disk	250
	2	1 x ATDT: 1 Dual Core Processor , 4GB RAM, 300GB Disk	500
Scale-up Servers	3	1 x AT: 1 Dual Processor, 4GB RAM, 500GB Disk	2200
	4	1 x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	
	5	1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk	3600
	6	1 x DT: 2 Quad Core Processor, 16GB RAM, 3TB Disk	
Scale-out Servers	7	n x AT: 1 Dual Processor, 4GB RAM, 500GB Disk m x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	3600+

Table 3 – Maximum recommended users

NOTE

Build Servers and Lab Management environments are outside the scope of this guidance. Refer to [Build Customization Guidance](#) ⁶ and [Lab Management Guidance](#) ⁷ respectively for guidance about Build Servers and Lab Management.

⁶ <http://go.microsoft.com/fwlink/?LinkID=230938>

⁷ <http://go.microsoft.com/fwlink/?LinkID=230951>

Planning Guide – Defining your TFS Strategy

Deployment Option	REF ⁸	Example specification	Physical Hardware Max users
Single Server	1	1 x ATDT: 1 Processor, 2GB RAM, 125GB Disk	250
	2	1 x ATDT: 1 Dual Core Processor , 4GB RAM, 300GB Disk	500
Scale-up Servers	3	1 x AT: 1 Dual Processor, 4GB RAM, 500GB Disk	2200
	4	1 x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	
	5	1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk	3600
	6	1 x DT: 2 Quad Core Processor, 16GB RAM, 3TB Disk	
Scale-out Servers	7	n x AT: 1 Dual Processor, 4GB RAM, 500GB Disk m x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	3600+ ⁹

Table 4 – Recommended maximum users for on-premises server deployments

NOTE

If you use **SharePoint 2010**, the installation of TFS will enforce a minimum of **4GB RAM** and raise a warning if the system has less than **10GB RAM**. This is a SharePoint, not a TFS constraint.

If you use **SharePoint 2013**, the SharePoint 2013 Distributor Cache which is a new feature which reserves 10% of RAM on any machine it get started at minimum . It's not recommended to install Distributor Cache on a Server Running SQL. Read more [here](#) ¹⁰.

Examples – Customer Types and Maximum Users

Humongous Insurance needs to support 500 TFS users. If this is the absolute maximum number of users, we could consider a single server, but considering a scale-up multi server deployment option would be recommended.

Trey Research needs to support 20 TFS users. Although the scale-up and scale-out deployment options are feasible, the single server deployment option is recommended and will provide considerable room for growth.

Consolidated Messenger needs to support 100 in-house users and 1000 other users. The scale-up multi server deployment option will deliver scalability up to three times the amount of their user limit, making it the recommended deployment option. Because of the distributed nature of teams and users, you should investigate the use of TFS Proxy Servers to cache copies of version control files in the location of distributed teams and thereby reduce bandwidth requirements.

See [Configure Team Foundation Version Control to use Proxy Server](#) ¹¹ for more details on the Proxy servers.

⁸ REF ... Reference numbers used by the Cheat Sheets when referring to these hardware specifications

⁹ Please note that there is a maximum number of users, calculated for AT and DT servers combined. You can scale up to 10,000 or more users as long as they are distributed across different AT and DT servers.

¹⁰ <http://blogs.technet.com/b/uktechnet/archive/2013/05/07/guest-post-distributed-cache-service-in-sharepoint-2013.aspx>

¹¹ <http://msdn.microsoft.com/en-us/library/ms245478.aspx>

Planning based on number of Team Project Collections – Option 3

The soft limit imposed by the number of active Team Project Collections is summarized in **Table 5**, whereby TFS can support from 10 (low end) to 8000 (high end) Team Project Collections per SQL Server instance:

Deployment Option	REF	Example specification	Max Active ¹² Team Project Collections ¹³
Single Server	1	1 x ATDT: 1 Processor, 2GB RAM, 125GB Disk	5
	2	1 x ATDT: 1 Dual Core Processor , 4GB RAM, 300GB Disk	10
Scale-up Servers	3	1 x AT: 1 Dual Processor, 4GB RAM, 500GB Disk 1 x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	75
	4	1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk 1 x DT: 2 Quad Core Processor, 16GB RAM, 3TB Disk	90
	5	1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk 1 x DT: 2 Quad Core Processor, 32 GB RAM, 3TB Disk	125
	6	1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk 1 x DT: 2 Quad Core Processor, 64 GB RAM, 3TB Disk	195
Scale-out Servers	7	n x AT: 1 Dual Processor, 4GB RAM, 500GB Disk m x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	75+

Table 5 – Recommended max active TPCs for on-premises server deployments

Refer to **Defining your Team Project Collection Strategy** on page 23 for more information about Team Project Collections.

Planning based on number of Team Projects – Option 4

The Team Project constraint is based on resource usage and impacts the overall system performance, both on the server and the client. The current limits for Team Projects based on the common process templates per Team Project Collection are:

Process Template	Max Team Projects ¹⁴
Agile	1000
CMMI	250
Scrum	1000

Table 6 – Recommended maximum Team Projects for on-premises server deployments

Considering Upgrade and Migration

When you are performing an upgrade of TFS you will want to consider increasing the server resources you have allocated. “Throwing everything you have at the environment” has been a term used in previous TFS upgrades. The TFS import process is resource intensive so there are several things to plan for and to keep in mind, especially if you have large (100GB+) version control databases:

¹² An active collection is one that will be accessed on a daily frequency.

¹³ Per SQL Server Instance

¹⁴ Rough estimate, which is dependent on modifications made to the process template.

- The number of processors that you can make available will help for a good portion of the upgrade process. The **import is multithreaded** with the exception of a few steps, so making at least eight processors available will be beneficial.
- Follow SQL Server best practices of having a **TempDB datafile per processor**. The upgrade process makes heavy use of TempDB for index creation and processing with Common Table Expressions (CTEs). Review the [Capacity for tempdb guidance](#) ¹⁵
- Follow SQL Server best practices of having the data, **log** and **TempDB** files on **separate drives/mountpoints**.
- Consider changing the default value MAXDOP setting in SQL Server if your installation kept the default of zero. Having a **MAXDOP equal to half of the number of processors** will benefit the multithreaded processing. Review the [Recommendations and Guidelines for "max degree of parallelism" configuration](#) ¹⁶ guidance.
- **Delete workspaces** that have not been used for a while. As a practice, Microsoft notifies workspace owners of a pending deletion of their workspace if it hasn't been used in the last six months. Builds create workspaces and you might be surprised to find how much space you could reclaim in your database by deleting unused workspaces.
- If you are not **defragmenting** your TFS **database(s)** and **re-indexing** them you should do this before you perform a migration. Optimized indexes will help when the migration is processing.
- **Pin SQL Server memory** to use a minimum of 65% and a maximum of 80%. You will see that the upgrade process will take the maximum amount of SQL memory at points during the processing.
- If you have a multi AT environment, remember that the upgrade process can run off of any of the TFS AT servers. In the case of running the import command on one server, it doesn't mean that steps will run exclusively from that server. You will want the servers in the multi AT environment to have the same resource allocation.
- You should have your **warehouse processing stopped** during the migration to conserve your SQL Server resources and reduce database contention.

Plan for several **trial runs** if your production development outage window is critical to your business. Your testing environment's resources should match your production environment where you will be performing your upgrade. You don't want any unpleasant surprises when performing your upgrade.

Check the **recovery mode** of your databases because the upgrade process will be generating a lot of transactions. You want to make sure that you don't run into problems with autogrows or transaction logs becoming full. The TFS data warehouse can be in simple recovery mode as well as the Reporting Services databases. By default, these are both in full recovery mode. If you are upgrading TFS 2008, you will want to check the transaction log file autogrow settings for the TFSVersionControl database. Because this database ends up being the upgraded TFS 2012 database, it is a good practice to at least double the size of the log file.

NOTE

You should consider doubling or quadrupling the resources allocated to your TFS environment for the upgrade processing. You can read more about common upgrade and migration scenarios in the [Visual Studio ALM Rangers Upgrade Guidance](#) ¹⁷.

Most environments are virtualized and adding/removing resources are relatively easy to do. You should consider doubling or quadrupling the resources allocated to your TFS environment for the upgrade processing. Once your TFS environment is live you can scale back on resources in a virtual environment to align more with the guidance identified earlier in this document.

¹⁵ <http://msdn.microsoft.com/en-us/library/ms345368.aspx>

¹⁶ <http://support.microsoft.com/kb/2023536>

¹⁷ <http://go.microsoft.com/fwlink/?LinkID=230948>

Considering Virtualization

You can virtualize most, if not all, of the TFS infrastructure. Refer to [Visual Studio ALM Rangers VM Factory Tooling and Guidance](#)¹⁸ for guidance on virtualization, tooling and reference solutions.

It is imperative that you validate and “dog food” your virtualized infrastructure before committing it to your production environment.

Rule of thumb for virtualization capacity planning

The rule of thumb for virtualization capacity planning is:

- Increase the recommended physical server hardware specifications by 15-20%
- or-
- Decrease the recommended maximum users by 15-20%

The following table shows the comparison of recommended maximum users for physical and virtual environments, based on the section **Planning based on number of users** on page 15.

Example – Reduce maximum users by 20% for virtualization

Deployment Option	Example specification	Physical Hardware Max users	Virtualized Hardware Max Users
Single Server	1 x ATDT: 1 Processor, 2GB RAM, 125GB Disk	20	16
	1 x ATDT: 1 Processor, 2GB RAM, 125GB Disk	250	200
	1 x ATDT: 1 Dual Core Processor , 4GB RAM, 300GB Disk	500	400
Scale-up Servers	1 x AT: 1 Dual Processor, 4GB RAM, 500GB Disk	2200	1760
	1 x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk		
	1 x AT: 1 Quad Core Processor , 8GB RAM, 500GB Disk 1 x DT: 2 Quad Core Processor, 16GB RAM, 3TB Disk	3600	2880
Scale-out Servers	n x AT: 1 Dual Processor, 4GB RAM, 500GB Disk m x DT: 1 Quad Core Processor, 8GB RAM, 2TB Disk	3600+	2880+

Table 7 – Recommended maximum users in virtualized environments

High Availability Strategies

Refer to the latest TFS Installation Guide¹⁹ for high availability options, which typically include a combination of multiple servers, a SQL Server cluster and a farm of Application tier servers.

¹⁸ <http://go.microsoft.com/fwlink/?LinkID=230953>

¹⁹ <http://www.microsoft.com/en-gb/download/details.aspx?id=29035>

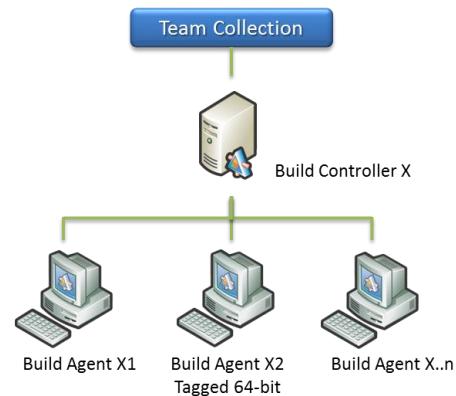
Build Server Considerations

Please refer to the Build Customization Guidance for details and guidance on Team Foundation Build Topology and the build process.

A general consideration should be that if you are using TFS you should have at least one build server.

Things to consider which might affect the number of build servers required:

- The **build machine** is the hardware that hosts the build service, which in turn can host a build controller service and/or build agents. Each build machine is dedicated to a given Team Project Collection and can host a build controller or a build agent, or both.
- Are there certain branches or Team Projects which require **specialized** and **incompatible versions**? If there are incompatibilities, the number of servers required increases.
- Are there '**bursts**' in requests? For example, do multiple teams all work to a certain milestone when builds are taken to deliver to test? If this is the case then more capacity might be needed to avoid bottlenecking and queuing.
 - The question on how much more capacity is needed is based on:
 - How long are the builds?
 - How long can the team wait for a build?
 - How much money can be thrown at the problem?
- Are there builds that have **dedicated build server times**? For example, build 1 generates the test environment data, builds 2 and 3 are noon bursts, with additional builds running in the afternoon, all of which could be dependent on each other. If you have this requirement, refactor and try and avoid this constraint.
- Are you after **cost effective** infrastructure, or after **efficient builds** which require decent hardware?
- Can you **consider virtualization of the build machine**? The Team Foundation Build Topology lends itself to virtualization!



Useful Reference Information

- [Understanding a Team Foundation Build System](http://msdn.microsoft.com/en-us/library/dd793166.aspx) ²⁰
- [Build System Examples](http://msdn.microsoft.com/en-us/library/dd793166.aspx#build_system_examples) ²¹
- [System Requirements for Team Foundation Build Service](http://msdn.microsoft.com/en-us/library/dd578619.aspx) ²²
- [Dogfooding Team Foundation Build: By The Numbers \(December 2011\)](http://blogs.msdn.com/b/willbar/archive/2012/01/08/dogfooding-team-foundation-build-by-the-numbers-december-2011.aspx) ²³

²⁰ <http://msdn.microsoft.com/en-us/library/dd793166.aspx>

²¹ http://msdn.microsoft.com/en-us/library/dd793166.aspx#build_system_examples

²² <http://msdn.microsoft.com/en-us/library/dd578619.aspx>

²³ <http://blogs.msdn.com/b/willbar/archive/2012/01/08/dogfooding-team-foundation-build-by-the-numbers-december-2011.aspx>

Proxy Server Considerations

The purpose of the TFS Proxy is to manage a cache of downloaded version control files. If you implement a TFS Proxy, you will download each source control version of a file one time per location instead of one time for every developer, which improves the download performance when developers are working on the **same** TFS Project or even more than one Project. This can be significant where long distances or poor connectivity impact how long it takes to download code.

This [post](#)²⁴ does a great job of explaining all of the communication that goes on between the Visual Studio workstation, TFS Proxy and Team Foundation Server. The TFS Proxy does not completely eliminate all traffic from the workstation to the Team Foundation Server. Authentication of the remote user is still performed against the Team Foundation Server so there is some communication going back and forth.

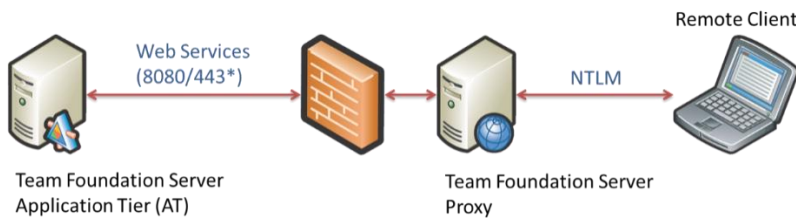


Figure 7 - TFS proxy

The default port of the TFS Proxy is **8081**. However, it can be changed and if SSL is needed it can be done over port **443**. Metrics to calculate the need for a Proxy Server is beyond the scope of this guide, but the math is pretty simple. Generally, if you have a remote team of more than 3-5 people, you should see improvements when you place a proxy server at their location. There should, however, be a significant geographic distance, or slow network connectivity between the Team Foundation Server and the TFS Proxy to see the value of implementing it.

NOTE

Don't place the TFS Proxy on the Team Foundation Server. This is **not** the correct configuration for implementing the proxy server.

Installing the TFS Proxy requires the TFS installation media. Connecting the TFS Proxy requires connectivity to the Team Foundation Server. To use TFS from the Visual Studio client, you have to set the TFS Proxy name and port in **Tools, Source Control, Visual Studio Team Foundation Server**.

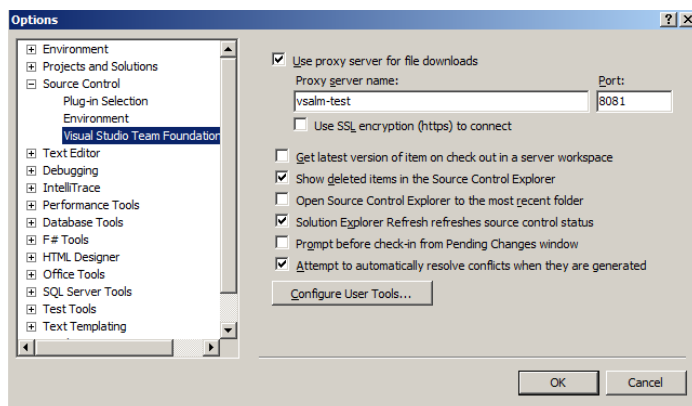


Figure 8 - Proxy Server configuration in Visual Studio

²⁴ <http://blogs.msdn.com/b/tsyang/archive/2008/03/24/how-team-foundation-server-proxy-2008-works.aspx>

Do not specify **http://** or **port** when configuring the Proxy server values in Visual Studio.

The value of the Proxy Server can be found in the TFS Admin Console. Use the URL field on the TFS Proxy page.



Figure 9- TFS proxy in administration console

Remember that the TFS Proxy is intended to improve the download experience. It will not make the check-in (upload) process any faster. To verify that developers are using the TFS Proxy you can check the IIS logs on the server. The logs include relevant information such as the IP address of the workstation, duration to download, username used to connect to the server and the version of Visual Studio that was used. Applications like [LogParser](#)²⁵ are beneficial for rolling up statistics of the IIS log.

```
2012-01-29 03:31:30 W3SVC4 VSALM-TEST GET /versionControlProxy/v1.0/item.aspx type=rsa&sfid=1365,1364,1367,1369,1366,1370,1368,1372,1373,1371&ts=634634910882094512&w=yy
2012-01-29 03:31:30 W3SVC4 VSALM-TEST GET /versionControlProxy/v1.0/item.aspx type=rsa&sfid=1365,1364,1367,1369,1366,1370,1368,1372,1373,1371&ts=634634910882094512&w=yy
2012-01-29 03:41:04 W3SVC4 VSALM-TEST ::1 GET /versionControlProxy/v1.0/ProxyStatistics.aspx - 8081 - ::1 HTTP/1.1 Mozilla/4.0+(compatible;+MSIE+8.0;+windows+NT+6.1;+wO
2012-01-29 03:41:05 W3SVC4 VSALM-TEST ::1 GET /versionControlProxy/v1.0/ProxyStatistics.aspx - 8081 VSALM-TEST\Administrator ::1 HTTP/1.1 Mozilla/4.0+(compatible;+MSIE+
2012-01-29 03:41:05 W3SVC4 VSALM-TEST ::1 GET /favicon.ico - 8081 VSALM-TEST\Administrator ::1 HTTP/1.1 Mozilla/4.0+(compatible;+MSIE+8.0;+windows+NT+6.1;+wOw64;+Triden
2012-01-29 03:41:08 W3SVC4 VSALM-TEST ::1 GET /versionControlProxy/v1.0/ProxyStatistics.aspx qp=QueryProxyStatistics 8081 VSALM-TEST\Administrator ::1 HTTP/1.1 Mozilla/
2012-01-29 03:41:08 W3SVC4 VSALM-TEST ::1 GET /favicon.ico - 8081 VSALM-TEST\Administrator ::1 HTTP/1.1 Mozilla/4.0+(compatible;+MSIE+8.0;+windows+NT+6.1;+wOw64;+Triden
2012-01-29 03:41:13 W3SVC4 VSALM-TEST ::1 POST /versionControlProxy/v1.0/ProxyStatistics.aspx/QueryProxyStatistics - 8081 - ::1 HTTP/1.1 Mozilla/4.0+(compatible;+MSIE+8
2012-01-29 03:41:13 W3SVC4 VSALM-TEST ::1 POST /versionControlProxy/v1.0/ProxyStatistics.aspx/QueryProxyStatistics - 8081 VSALM-TEST\Administrator ::1 HTTP/1.1 Mozilla/
```

Figure 10 - IIS Log showing TFS proxy traffic

REFERENCES

- [Visual Studio Team Foundation Service Guidance](#)²⁶
- [Visual Studio ALM Rangers Quick Reference Guidance](#)²⁷
- [Visual Studio ALM Rangers VM Factory Tooling and Guidance](#)²⁸
- [Visual Studio ALM Rangers Upgrade Guidance](#)²⁹
- [Team Foundation Server 2012 Installation Guide](#)³⁰
- [Team Foundation Server 2012 Administration Guide](#)³¹
- [Team Foundation Server Architecture](#)³²
- [Team Foundation Server 2010 Installation Guide](#)³³
- [Team Foundation Server 2010 Administration Guide](#)³⁴
- [Professional Team Foundation Server](#)³⁵ book
- [Professional Application Lifecycle Management with Visual Studio 2010](#)³⁶ book
- [Team Development using Microsoft Visual Studio Team Foundation Server](#) P&P³⁷ book

²⁵ <http://support.microsoft.com/kb/910447>

²⁶ <http://go.microsoft.com/fwlink/?LinkID=230945>

²⁷ <http://vs2010quickref.codeplex.com/>

²⁸ <http://go.microsoft.com/fwlink/?LinkID=230953>

²⁹ <http://go.microsoft.com/fwlink/?LinkID=230948>

³⁰ <http://go.microsoft.com/fwlink/?LinkID=231279>

³¹ <http://go.microsoft.com/fwlink/?LinkID=231280>

³² <http://msdn.microsoft.com/en-us/library/ms252473.aspx>

³³ <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24337>

³⁴ <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=8175>

³⁵ <http://www.amazon.com/Professional-Team-Foundation-Server-Programmer/dp/0470943327>

³⁶ <http://www.amazon.com/Professional-Application-Lifecycle-Management-Programmer/dp/0470484268>

³⁷ <http://www.amazon.com/gp/product/0735625719>

Defining your Team Project Collection Strategy

This section introduces you to **Team Project Collections** and covers a few scenarios which will assist you in deciding whether you need one or more Team Project Collections.

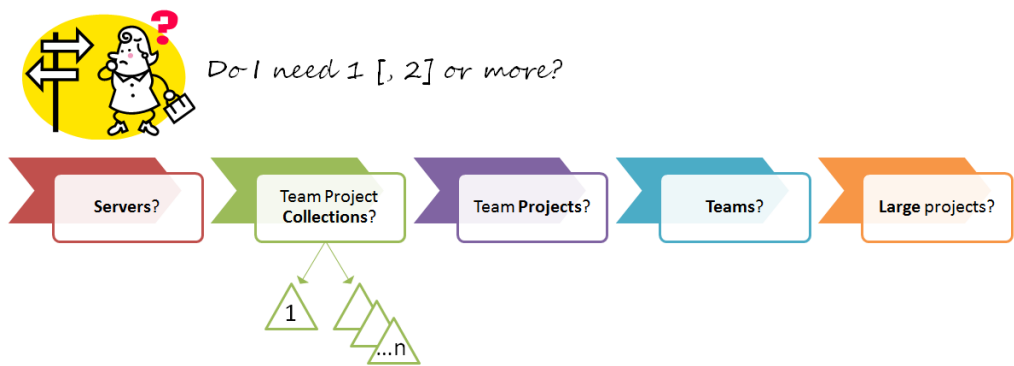


Figure 11 - Team Project Collection planning

Refer to Hands-on Lab, section “*Stepping through the planning of a Team Project Collection Strategy,*” for a walkthrough of this section.

This section is focused on **Dave** the TFS administrator and **Garry** the lead for development teams.



I would like to ...	Page(s)
• Understand Team Project Collections	24
• Understand Team Project Collection Backup strategies	24
• Understand Team Project Collection Security Isolation	25
• Understand Team Project Collection Constraints	25
• Deciding on a Team Project Collection strategy	26

Table 8 - Deciding what information is important with Team Project Collection planning

Understanding Team Project Collections

Overview

TFS 2010 introduced the feature of Team Project Collections. Each TFS instance can host one or more Team Project Collections, each of which can accommodate one or more Team Projects. Similar to a SharePoint Collection, the Team Project Collection acts as a container and as a basic unit of isolation and archiving.

For more information about Team Project Collections and management thereof, please refer to [Organizing Your Server with Team Project Collections](#)³⁸.

Isolation Considerations

The acts of isolating Team Projects in Team Project Collections are concerned with:

- Scalability
- Backup and Restore
- Recovery
- Security isolation

Scalability

TFS supports the ability to distribute Team Project Collection databases across multiple SQL Server Instances, providing an effective scale out solution for your deployment.

Backup, Restore and Recovery

Team Project Collections are the basic unit of recovery for TFS, allowing all Team Projects contained within a Team Project Collection to be backed-up and or restored as a unit.

NOTE

For the purpose of backup and recovery, consider isolating Team Projects into separate Team Project Collections. For example, if you have a requirement to have individual backup and restorable units for a Team Project the only scenario to accomplish this is to have one Team Project per collection.

For more information and an example scenario for archiving, please refer to [Visual Studio TFS Team Project and Collection Guidance](#)³⁹.

For more information about splitting and merging Team Project Collections, please refer to Visual Studio ALM Rangers [Team Foundation Server Upgrade Guidance](#)⁴⁰.

³⁸ <http://msdn.microsoft.com/en-us/library/dd236915.aspx>

³⁹ <http://msdn.microsoft.com/en-us/magazine/gg983486.aspx>

⁴⁰ <http://go.microsoft.com/fwlink/?LinkID=230948>

Security Isolation

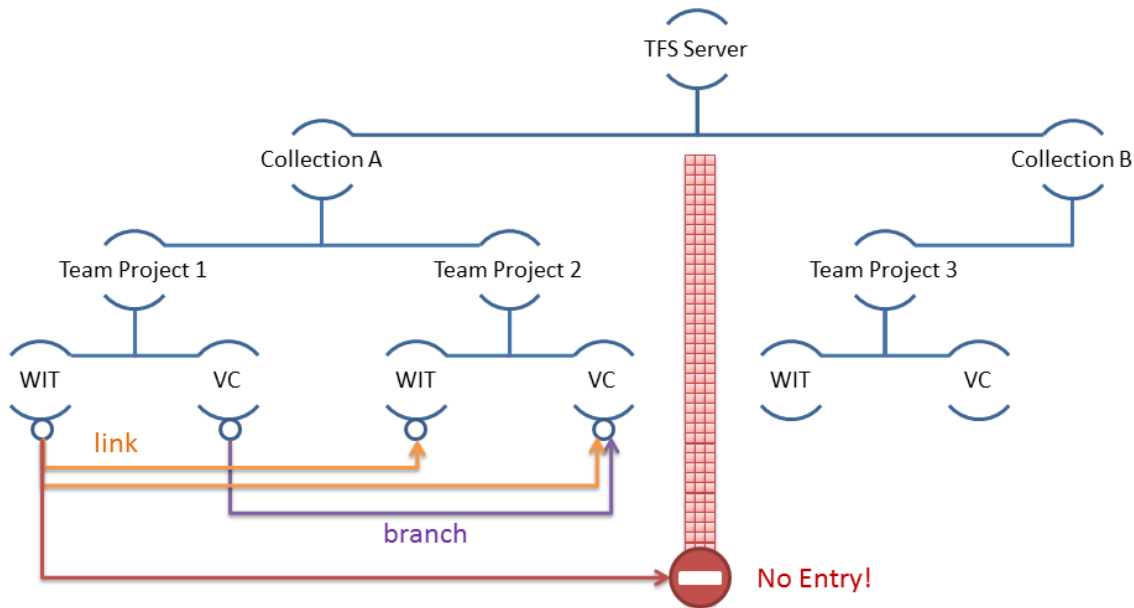


Figure 12 - Team project collection sharing and isolation boundaries

As shown in the above illustration, sharing of Team Project artifacts among Team Projects is possible within the same Team Project Collection. There is, however, an isolation boundary between Team Project Collections and artifacts cannot be shared across Team Project Collection boundaries. If Team Projects must share artifacts, they must be contained with the same Team Project Collection.

Team Project Collection Constraints

The following table summarizes the constraints associated with Team Project Collections:

Feature	Constraint
# Active collections	30-100 per SQL Server instance
# Total Collections	10-8000 per SQL Server instance
Branching of Version Control	Not allowed across Team Project Collections
Build Machine Scoping	Scoped to one Team Project Collection
Connecting as user	To one Team Project Collection at a time
Renaming	Not supported to rename Team Project Collections
Sharing of artifacts	Not allowed between Team Project Collections
Work Item Linking	Not allowed across Team Project Collections
Work Item Queries	Not allowed across Team Project Collections

Table 9 - Team project collection constraints

Key Decisions – Single versus Multiple Team Project Collection Strategy

 Feasible
  Limitations
  Not feasible



























	(... in different Team Project Collections)	Single	Multiple
Sharing	Work Items between Team Projects		
	Source Code between Team Projects		
	Queries between Team Projects		
	Reports between Team Projects		 As we have one analysis database we can, in theory, have reports between Team Projects within one deployment.
	Users between Team Projects		
Usage	Categorization of Team Projects for navigation ⁴¹	 Categorization of Team Projects within a single Team Project Collection is limited on naming and sorting as it is a flat list.	
	Switching between Team Projects		 Switching between Team Projects in different Team Project Collections is possible, but requires a re-connect of Team Explorer which is a time consuming exercise.
Management	Scalability		
	Isolation between Team Projects	 Isolation between Team Projects within the same Team Project Collection can be achieved with security isolation, but is a costly administrative task.	
	Different Team Project Collection security model		
Customer Types	Humongous Insurance		
	Trey Research		
	Consolidated Messenger	 Consolidated Messenger customer types may require additional scalability and fault tolerance, which may require load balancing of Team Projects across multiple Team Project Collections.	

Table 10 – Key Decisions – Single versus multiple Team Project Collection strategy

Deciding on Team Project Collection Strategy

We recommend keeping the number of Team Project Collections as small as possible to maximize simplicity and minimize administration. When you have a good reason to add an additional Team Project Collection, primarily to support isolation and scalability requirements, the pros and cons that are outlined on pages 24-25 start to matter.

⁴¹ Refers to ability to categorize Team Projects, which are currently represented in a flat and sorted list of Team Projects within a Team Project Collection.

Consider giving the Team Project Collection a descriptive name, as renaming is not supported and the default name of “**DefaultCollection**” might not refer to the default Team Project Collection in future.

Customer Types and Team Project Collections

The single Team Project Collection strategy is suitable for both Humongous Insurance and Trey Research because neither has a real requirement for isolation of Team Projects or scalability at this stage.

Consolidated Messenger could consider the single Team Project Collection strategy, addressing isolation requirements between customer focused Team Projects using granular access security. Considering its current size and the potential for growth, however, the customer type leans towards a scale-out infrastructure, which in turn introduces the ability to load balance multiple Team Project Collections amongst two or more SQL Servers.

New, Old and Test Team Project Collection Strategy

When you are dealing with short-term evaluation and test Team Projects, Team Projects created by migrating from other environments (Rational, Visual SourceSafe, Perforce, Subversion, etc.) and new Team Projects, you might want to split up the Team Projects into three or more Team Project Collections—one for testing only, one for reference of migrated projects and one or more for new projects.

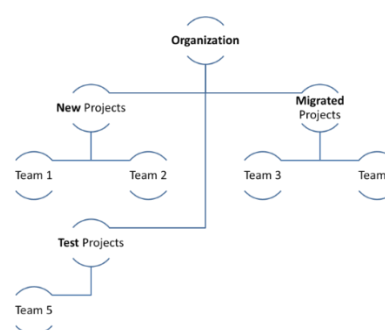


Figure 13 - New, old and test tpc strategy

Single Team Project Collection Strategy

In scenarios where isolation and scalability is not a consideration, the single Team Project Collection is the appropriate strategy for an organization and team to adopt. The move to multiple Team Project Collections is straight forward if the need arises, whereby the product documentation on MSDN and the [Team Foundation Server Upgrade Guidance](http://go.microsoft.com/fwlink/?LinkID=230948)⁴² cover the possible splitting and merging of Team Project Collections that might emerge with such a move.

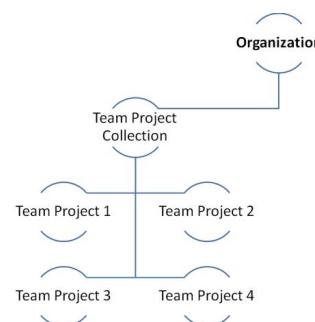


Figure 14 - Single Team Project Collection strategy

As shown in Figure 14, the **single** Team Project Collection scenario implements one Team Project Collection, which is similar to a SharePoint Site Collection in terms of isolation and basic unit of recovery.

The advantages you have to consider are:

- **Simplicity** for users
 - No need to switch context between Team Project Collections
 - No need to search for Team Projects in different Team Project Collections

⁴² <http://go.microsoft.com/fwlink/?LinkID=230948>

- Ability to **share artifacts** between Team Projects
- **Single database** for all Team Project data

The disadvantages you have to consider are:

- **Security complexity** and **administration** needed to achieve isolation within a Team Project Collection
- As the number of active Team Projects grows, the flat list within a Team Project Collection could make **navigation** between Team Projects **difficult**.

Multiple Team Project Collections Strategy

Concepts

When you need to enforce strict isolation between Team Projects, or when you need to consider load balancing and scalability, consider the **multiple** Team Project Collection, as shown in **Figure 16**.

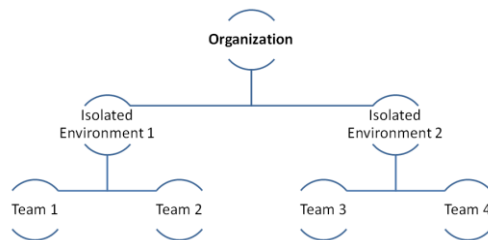


Figure 15 – Organizational isolation

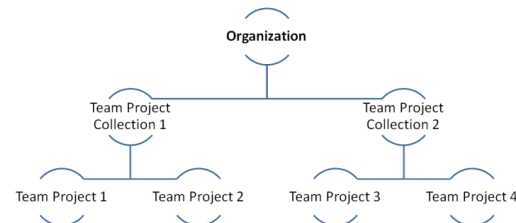


Figure 16 - Multiple Team Project Collection strategy

Organizational Structure and possible Team Project Collection correlations

As shown in **Figure 15** and **Figure 16** the multiple Team Project Collection scenarios allow you to represent an organizational structure with a similar looking Team Project Collection and Team Project model.

The advantages you have to consider are:

- You can implement **isolation strategies** (project, product and organization)
- You can implement different **security isolation** strategies per Team Project Collection
- You can **manage** (start, stop, backup) Team Project Collections individually
- You can **distribute** Team Project Collection databases across one or more SQL Server instances

The disadvantages you have to consider are:

- You **cannot** share **artifacts**, such as work items, source code and queries, cross Team Project Collections
- You need to **switch** among two or more Team Project Collections, which can become unproductive for teams working in many Team Project Collections.
- You need to implement multiple Build Controllers as they are scoped to a Team project Collection.

References

- Visual Studio ALM Rangers [Quick Reference Guidance](http://vs2010quickref.codeplex.com/) ⁴³
- Visual Studio ALM Rangers [Team Foundation Server Upgrade Guidance](http://go.microsoft.com/fwlink/?LinkID=230948) ⁴⁴
- MSDN Magazine: [Visual Studio TFS Team Project and Collection Guidance](http://msdn.microsoft.com/en-us/magazine/gg983486.aspx) ⁴⁵
- [Organizing Your Server with Team Project Collections](http://msdn.microsoft.com/en-us/library/dd236915.aspx) ⁴⁶

⁴³ <http://vs2010quickref.codeplex.com/>

⁴⁴ <http://go.microsoft.com/fwlink/?LinkID=230948>

⁴⁵ <http://msdn.microsoft.com/en-us/magazine/gg983486.aspx>

⁴⁶ <http://msdn.microsoft.com/en-us/library/dd236915.aspx>

Defining your Team Project Strategy

This section introduces you to **Team Projects** and covers a few scenarios that will assist you in deciding whether you need one or more Team Projects.

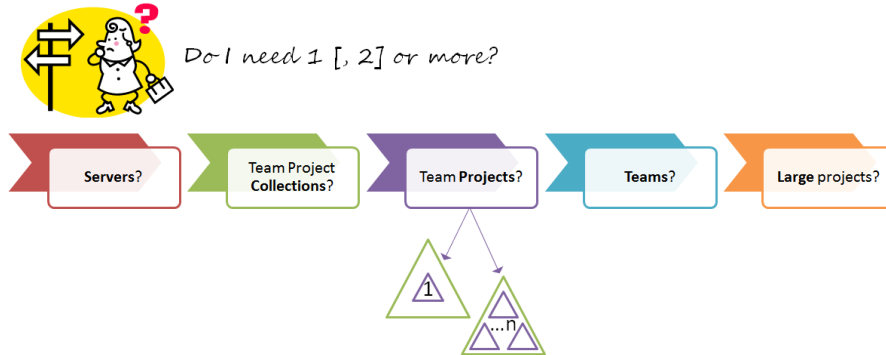


Figure 17 - Team Projects planning

Refer to Hands-on Lab, section ***"Stepping through the planning of a Team Project Strategy"*** for a walkthrough of this section.

This section is focused on **Dave** the TFS administrator, and **Garry** the lead for development teams.



I would like to ...	Page(s)
• Understand Team Projects	24
• Understand Team Project Boundaries	31
• Understand Team Project Constraints	31
• Deciding on a Team Project strategy	31

Table 11 - Deciding what information is important with Team Project planning

Understanding Team Projects

Before you can get started with inserting data into TFS, you have to create one or more Team Projects within a Team Project Collection. Having said this, you can imagine a Team Project as some kind of container for related data within a defined scope. Even though the term “Team Project” implies that the scope of a team should somehow relate to a team of developers, that’s not necessarily the case. Several approaches exist how to define the scope of a Team Project, which will be covered in the next chapter. Before we get there, this chapter will clarify the physical aspects of a Team Project.

Ingredients of a Team Project

A Team Project consists of related data. All the data of a Team Project is stored within one single Team Project Collection and therefore in a single database. Therefore, the main purpose of working with multiple Team Projects within a single Team Project Collection is rather a logical isolation of data than a physical isolation.

Some of the data is defined when a Team Project is created, some can be modified during the lifecycle of a Team Project, some can be shared between Team Projects (see next chapter) but all of the data strictly belongs to a single Team Project. Most of the data of a Team Project can be accessed using Team Explorer.

NOTE

Actually it’s not true, that **all** of the data of a Team Project is stored in a single database. Some of the data is related to a single Team Project, but still distributed over several databases, for example, Documents in SharePoint Portal or reports. For the sake of simplicity we skip these exceptions in this topic.

The following data is being defined by a Team Project and therefore logically isolated from other Team Projects.

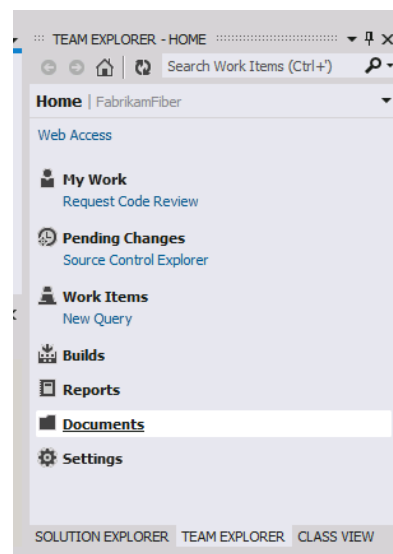
Team Project Name

The Team Project name has to be unique within a Team Project Collection. It cannot be changed later, so choose wisely.

Process

Every Team Project is based on a process template. Depending on the process template that you use, certain data, which defines the Team Project process infrastructure, is created and instantiated within a Team Project, this includes:

- Work Item Categories
- Work Item Types
- Link Types
- Queries
- Areas
- Iterations
- Microsoft Project Mapping Settings
- Groups & Permissions
- Lab Templates
- Build Templates
- Source-Control Settings
- Portal Settings
- Reports



NOTE

Later changes of the **base** process template will not affect existing instances of Team Projects, but every aspect defined by a process template can be changed during the lifecycle of a Team Project. See [TFS Process Template Customization Guidance](#)⁴⁷ for details. This is also true for the other direction: Changes made to an instance of a Team Project never effect on the base process template.

Members

Within a Team Project, you define team members to allow them to work on the Team Project with different permissions. For example, you can create new groups on Team Project level.

Work Items

Whenever you create a work item to track work, requirements or any kind of activity, you assign the work item to a single Team Project. Once assigned, it can never be moved to another Team Project.

Version Control

When you are working with one of the default process templates, several initial version control items will exist when you create a Team Project. The source code folder allows you to do all the source code activities that you are used to, like branching and merging, check-in and check-out, etc. A source code folder always belongs to a single Team Project.

Build Definitions

Build Definitions are a set of parameters which are used in combination with a certain build process template. Build Definitions are defined on a Team Project Level.

SharePoint Portal

Optional you can specify a SharePoint portal for a Team Project. This allows you to display data stored in SharePoint in a Team Project node in Team Explorer. Typically, a SharePoint Portal relates to a single Team Project.

Besides these rather physical aspects of a Team Project, a Team Project also defines logical settings. These include permissions for certain actions within the scope of the Team Projects, permissions to view or edit certain data or permissions for manipulation of permissions.

Team Project Boundaries

Some of the data of a Team Project can be shared among Team Projects; some can even be modified from other Team projects.

The following image illustrates the boundaries of Team Project inside a Team Project Collection and cross Team Project Collections (not recommended).

⁴⁷ <http://vsartfsptguide.codeplex.com/>

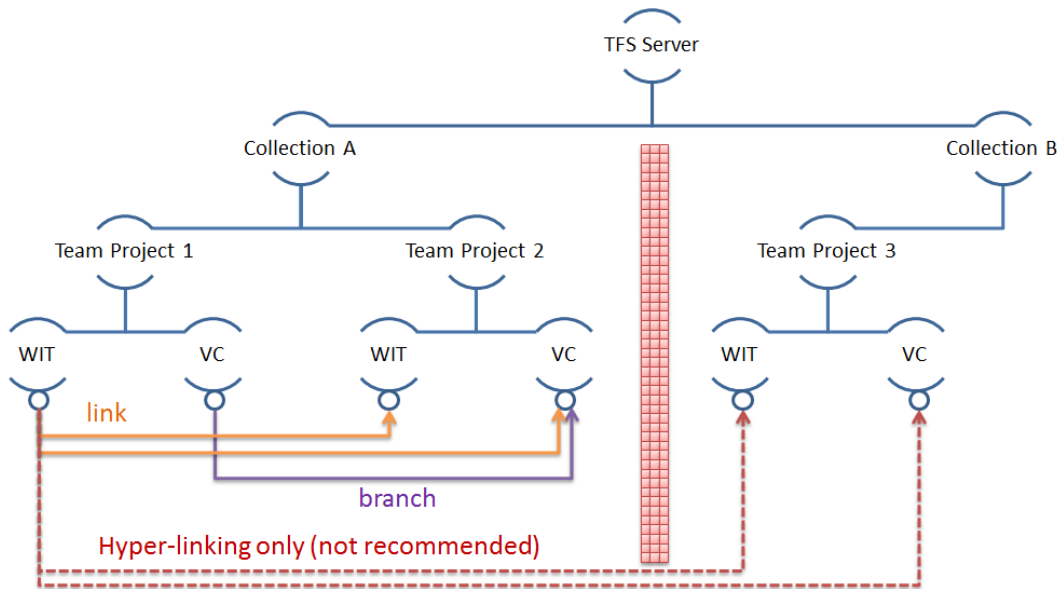


Figure 18 - Team project boundaries

Team Project Constraints

The following table summarizes the constraints associated with Team Projects:

Feature	Constraint
# Team Projects within a TPC	~ 1000 using Team Projects for MSF for Agile Software Development process template ~ 250 Team Projects for MSF for CMMI Process Improvement process template
Areas & Iterations	Cannot be shared among Team Projects
Base Process Templates	Shared within a Team Project Collection
Branching of Version Control	Allowed across Team Projects of the same collection
Build Definitions	Cannot be shared across Team Projects
Build Machine Scoping	Shared among Team Projects of a same Team Project Collection PC
Connecting as user	To multiple Team Projects within one Team Project Collection at a time
Global Lists	Shared within a Team Project Collection
Groups and Permissions	Cannot share permission settings using groups defined on Team Project level
Link Type Definitions	Defined on Team Project Collections level, can be shared among Team Projects
Source Control Settings	Shared among Team Projects of a same Team Project Collection
Team Project Name	Cannot be changed and is immutable
Work Item Categories	Defined on Team Project level, cannot be shared among Team Projects
Work Item Linking	Allowed across Team Projects of the same collection Across Team Project Collections only possible with usage of hyperlinks
Work Item Query Definitions	Not allowed across Team Project Collections, but possible across Team Projects of the same Team Project Collections.
Work Item Query Results	Can display data of several Team Projects of the same Team Project Collections
Work Item Templates	Cannot be shared among Team Projects
Work Item Type Definitions	Cannot be shared among instances of Team Projects

Table 12 - Team project constraints

Key Decisions – Single versus Multiple Team Project Strategy

 Feasible
 Limitations
 Not feasible































(... in the same Team Project Collection)		Single	Multiple
Sharing	Work Items between Team Projects		 Sharing of Work Items between Team projects may be restricted by security isolation.
	Source Code between Team Projects		 Sharing of Source Code between Team projects may be restricted by security isolation.
	Queries between Team Projects		
	Work Item Query Definitions between Team Projects		
	Work Item Query Results between Team Projects		
	Work Item Templates between Team Projects		
	Work Item Type Definitions between Team Projects		
	Reports between Team Projects		
Usage	Users between Team Projects		
	Categorization of Team Projects for navigation	 If you have one Team Project it becomes huge and hard to navigate between different projects/teams, similar to one Team Project Collection and many Team Projects.	
	Switching between Team Projects		 Switching between Team Projects may be restricted by security isolation.
Management	Different Team Project security model		
Customer Types	Humongous Insurance	 If Humungous Insurance supports several independent solutions, the single Team Project may become complex to navigate and administrate.	
	Trey Research		 Multiple Team projects may introduce unnecessary source code branching and merging administration to achieve haring of a base solution.
	Consolidated Messenger	 Single Team project may introduce unnecessary security isolation and navigation complexity.	

Table 13 – Key Decisions – Single versus Multiple Team Project Strategy

Deciding on a Team Project Strategy

NOTE

Please see the [Requirements Management for Ranger Projects – What I wish we did differently with our Team Projects](http://blogs.msdn.com/b/willy-peter_schaub/archive/2011/11/18/requirements-management-for-ranger-projects-what-i-wish-we-did-differently-with-our-team-projects.aspx)⁴⁸ blog post, which summarizes how we designed the ALM Rangers Team Project strategy and how we would like to change it ... unfortunately these designs are the immutable ones the teams have started working with.

Customer Types and Single Team Projects

The correlation between customer types and Team Project strategies are varied and the answer is probably always going to be “it depends.” Here are possible examples for the three known customer types:

- **Trey Research** might decide for the Single Team Project strategy, if all of the specialized solutions for numerous clients are based on the same code, with only slight modifications to the base solution.

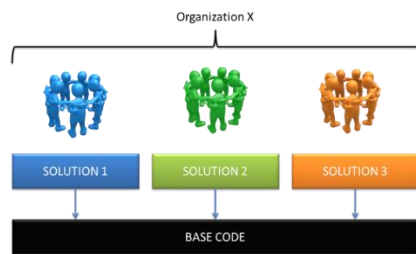


Figure 19 - Same code base

- **Humongous Insurance** might decide for the Multiple Team Project strategy, if they are developing several independent solutions.

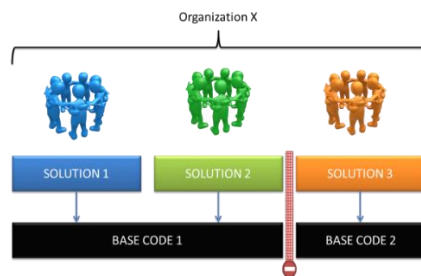


Figure 20 - Separate code base

- **Consolidated Messenger** might decide for the Multiple Team Project strategy if they have several development groups developing solutions for several customers.

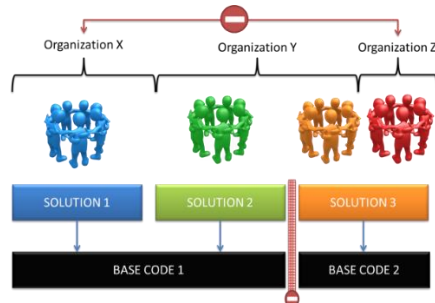


Figure 21 – Separate code base and dev groups

⁴⁸ http://blogs.msdn.com/b/willy-peter_schaub/archive/2011/11/18/requirements-management-for-ranger-projects-what-i-wish-we-did-differently-with-our-team-projects.aspx

Single Team Project Strategy

Concepts

Following the Single Team Project Strategy, as shown in **Figure 22**, you store all your data inside a single Team Project but create a substructure within the Team Project to isolate information inside the Team Project.

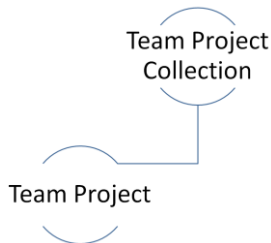


Figure 22 - Single Team Project strategy

Reasons for creating a single Team Project

Working with a single Team Project might be a good choice for you if your company and your development teams fulfill the following requirements:

- You do not find any good reasons to create another Team Project.
- You follow a single development process and all participants can agree on and follow the same TFS Process Template.
- You do not need logical isolation.
- You are working on a single component or product.
- You are working on multiple components or products but you are expecting intense sharing of code and information.

Downsides

On the downside, the Single Team Project Strategy might introduce:

- Finding other ways to sub-structure the work items of your Team Project
 - In the past, usage of the hierarchical Attribute "Area Path" was very common to structure your work items, so "Area Path" could be used for structuring into components.
 - Starting with TFS 2012, the "Area Path" is meant to be first choice to assign work items to a specific team and you could use another attribute to structure work items into components instead.
 - **Challenge:** This leads to a higher complexity in queries, reports etc.
- Finding other ways to structure your source code.
 - Depending on what you are developing, you might need to create additional subdirectories or branches, which might lead to a deeper hierarchies and longer file paths.
 - **Challenge:** This leads to a higher complexity with branching and merging, which requires a visit of the [Branching and Merging Guidance](#) ⁴⁹.
- Permissions must be set inside of the Team Project (Source code, Work items, Builds, Reports...) to control who sees and who can control the Team Project data.

⁴⁹ <http://go.microsoft.com/fwlink/?LinkID=230936>

Multiple Team Projects Strategy

Concepts

When you follow the Multiple Team Projects strategy, as shown in **Figure 23**, you create multiple Team Projects to store information isolated from each other. To find out about the level of isolation refer to page 31.

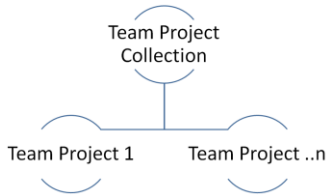


Figure 23 - Multiple Team Project strategy

Whenever you have multiple teams that are about to follow different processes, there is no other reasonable way than creating a new Team Project per process. Besides this, working with multiple Team Projects might be a good choice for you if your company and development team fulfill the following requirements:

- You understand the Single Team Project Strategy, see page 35, but don't feel 100% happy with it.
- You want a higher level of (logical) isolation.

Indicators

There are some rudimentary indicators which can help you to decide if a multiple project strategy is feasible:

- Your products or projects have **different** release cycles.
- Your products or projects have **different** stakeholders.
- Your products or projects have **different** source code base and technologies.
- Your products or projects have **different** domain relation.
- A single Team Project would be too big and introduce unproductivity and discomfort for users.

None of these indicators are hard criteria, but they can be indicators whether you should create a new Team Project.

Common examples for the usage of multiple Team Projects are:

- **Consolidated Messenger**

As a consulting company create one Team Project per customer, another Team Project for internal tool development:

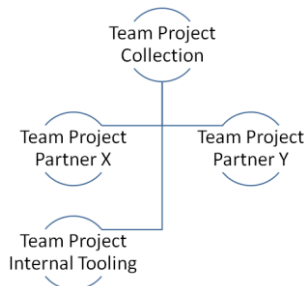


Figure 24 – Multiple Team Projects: different customers

- **Trey Research**

As an ISV, create one Team Project per product, another Team Project for base components, another Team Project for internal tool development:

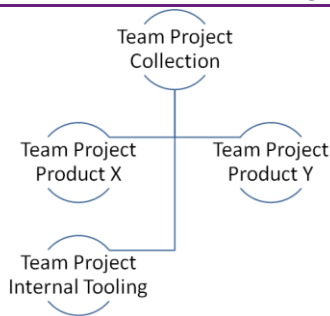


Figure 25 - Multiple Team Projects: different products

- **Humongous Insurance**

Create one Team Project per product per version:

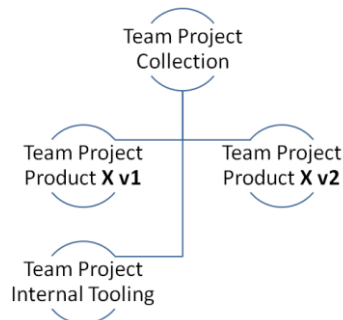


Figure 26 - Multiple Team Projects: different versions

Downsides

The downside of using multiple Team Projects is that you are losing comfort in some aspects:

- You have to find the “right” Team Project, to find and retrieve the information you need.
- You have to keep process templates in sync if you still want all your teams to work with the same process template.

References

- [Visual Studio ALM Rangers Quick Reference Guidance](http://vs2010quickref.codeplex.com/) ⁵⁰
- [Visual Studio ALM Rangers Branching and Merging Guidance](http://go.microsoft.com/fwlink/?LinkId=230936) ⁵¹
- MSDN Magazine: [Visual Studio TFS Team Project and Collection Guidance](http://msdn.microsoft.com/en-us/magazine/gg983486.aspx) ⁵²
- [Planning a Team Project](http://msdn.microsoft.com/en-US/library/ms242894(v=VS.80).aspx) ⁵³
- [Project Management](http://msdn.microsoft.com/en-us/library/bb668942.asp) ⁵⁴
- [Good Reasons to not create a new Team Project](http://msmvps.com/blogs/vstsblog/archive/2010/11.aspx) ⁵⁵
- [Planning a Team Project \(TFS 2008\)](http://msdn.microsoft.com/en-us/library/ms242894(v=vs.90).aspx) ⁵⁶
- [Creation of a Team Project](http://msdn.microsoft.com/en-us/library/dd286491.aspx) ⁵⁷

⁵⁰ <http://vs2010quickref.codeplex.com/>

⁵¹ <http://go.microsoft.com/fwlink/?LinkId=230936>

⁵² <http://msdn.microsoft.com/en-us/magazine/gg983486.aspx>

⁵³ [http://msdn.microsoft.com/en-US/library/ms242894\(v=VS.80\).aspx](http://msdn.microsoft.com/en-US/library/ms242894(v=VS.80).aspx)

⁵⁴ <http://msdn.microsoft.com/en-us/library/bb668942.asp>

⁵⁵ <http://msmvps.com/blogs/vstsblog/archive/2010/11.aspx>

⁵⁶ [http://msdn.microsoft.com/en-us/library/ms242894\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms242894(v=vs.90).aspx)

⁵⁷ <http://msdn.microsoft.com/en-us/library/dd286491.aspx>

Defining your Team Strategy

This section introduces you to **Teams** and covers a few scenarios which will assist you in deciding in how you can apply Teams to you development teams and whether you need one or more Teams.

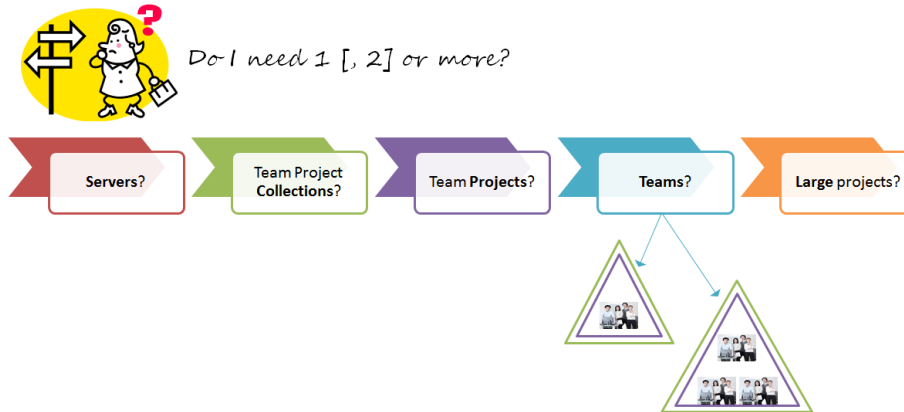


Figure 27 - Team planning

Refer to Hands-on Lab, section ***“Stepping through the planning of a Team Strategy”*** for a walkthrough of this section

This section is focused on **Dave** the TFS administrator, **Mike** the Program Manager, **Garry** the lead for development team, **Doris** the developer, **Christine** the tester and **Paul** the database administrator.



I would like to ...	Page(s)
• Understand Teams	39
• Understand Team Boundaries	39
• Understand Team Constraints	39
• Key Decisions – Single versus Multiple Team strategy	39
• Deciding on a Team strategy	40

Table 14 - Deciding what information is important with team planning

Understanding Teams

- Teams are new to TFS 2012 and they are defined within a Team Project. The Teams concept grew out of the need of Scrum teams.
- Teams allow you to set up a team that has their own Product Backlog to manage from the Team Project's Product Backlog.
- Teams manage their own Area Paths in a Team Project and become a collaboration hub for the team.
- The Team ends up being a named group of team members who work on a demarcated area of the Team Project.
- Every Team Project will have a default team created called My Team.

Team Boundaries

The Teams are constrained to their Team Project, with no ability to share data or teams among Team Projects. The following image illustrates the boundaries of Teams within Team Projects.

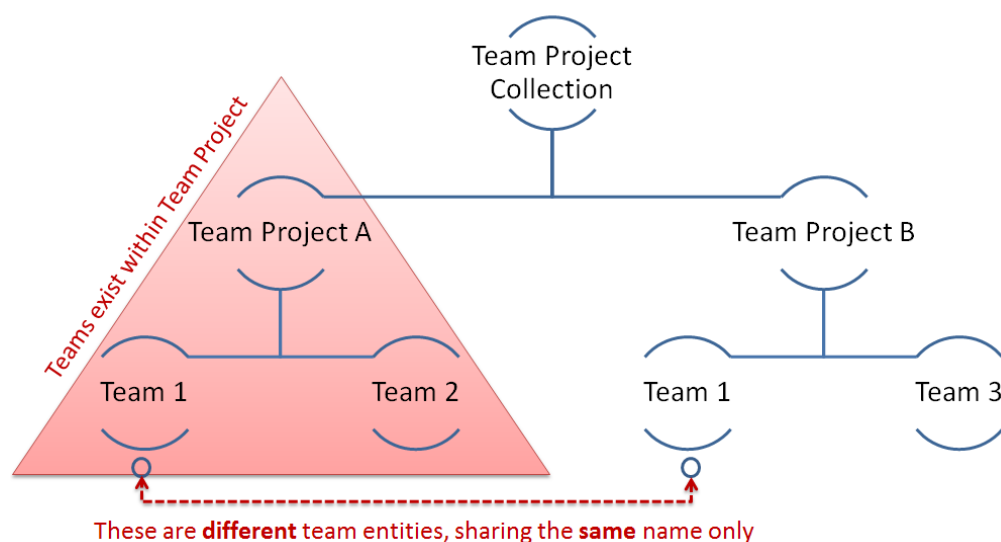


Figure 28 - Team boundaries

Team Constraints

The following table summarizes the constraints associated with Teams:

Feature	Constraint
Number of Team Members	<ul style="list-style-type: none"> • There is no hard limit of users within a Team. • Large groups are hard to navigate and manage. Try to create smaller sub teams if your team is large. • Groups larger than 100 will typically not be loaded by Visual Studio Team Explorer.
Team Name	<ul style="list-style-type: none"> • A team name must be unique within a Team Project. • "Team X" can exist in multiple Team Projects, but each will be a different team, sharing the same name.
Teams within Teams	Teams are a flat list of users and TFS Groups.

Table 15 - Team constraints

Key Decisions – Single versus Multiple Team Strategy

 Feasible
  Limitations
  Not feasible



























(... in the same Team Project)		Single	Multiple
Sharing	Work Items between Teams		
	Source Code between Teams		
	Queries between Teams		
	Reports between Teams		
	Users between Teams		
	Backlogs between Teams		
	Teams between Team Projects		
Usage	Categorization of different backlogs	 <p>If you have one Team it becomes huge and hard to navigate between backlog items and tasks.</p>	
	Grouping of team members	 <p>Groups larger than 100 will typically not be loaded by Visual Studio Team Explorer</p>	
	Switching between Teams		
Customer Types	Humongous Insurance	 <p>If Humongous Insurance supports several independent solutions, the single Teams may become complex to navigate and administrate.</p>	
	Trey Research		 <p>Multiple Team projects may introduce unnecessary administration.</p>
	Consolidated Messenger	 <p>Single Team project may introduce navigation and administration complexity with large teams.</p>	

Table 16 – Key Decisions – Single versus multiple team strategy

Area Paths and Teams

With TFS 2012, a team will own an area path and iteration. By default when a Team is created a Team area path will be created for the Team. Development teams may be using the area path to identify components in prior versions of TFS and have a custom field for tracking Teams that work on the component. The custom field created for tracking the Team can be tied into the Team functionality with TFS 2012 and the benefits of the use of teams will be transparent.

Deciding on Team Strategy

Single Team Strategy

Teams allow a team to manage the Product Backlog to a level that is more conducive to a team environment. If you have a single application in a Team Project and a relatively small team (fewer than 10 team members) then you can use one Team. We use the number 10 because the guidance for Scrum teams is to have nine or fewer team members. If you have 10 or more members on a Team, consider having them focus on particular functionality of an application. In this case you would have a good opportunity to use Teams to manage the capacity of the team and let the teams choose what Product Backlog items they are working on in their release.

Example

Project Background

The Trey Research TFS Project is a web project supported by two developers (Doris A. and Doris B.), tested by one QA person (Christine) and gets database support (Paul) on an as needed basis. The Trey Research Team Project is in maintenance mode because their code base has been stable and the only changes required to it are minor enhancements or bug fixes.

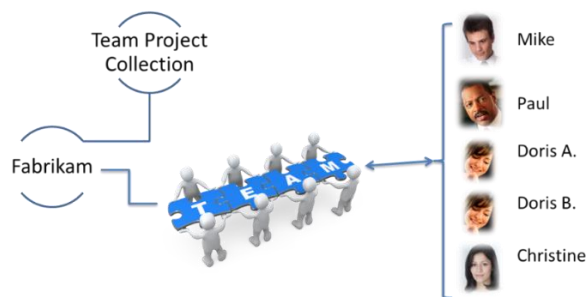


Figure 29 - Trey Research single team strategy

Using the Trey Research Team Project as a single Team approach, Paul, Doris A, Doris B. and Christine can be assigned as members on the Trey Research Team. Mike, the Program Manager, will create the tasks in the Product Backlog and define the sprints. Preliminary assignments will be made by Mike with the Team members capacity kept in mind. In this scenario, the number of team members is a manageable number and having multiple teams would add extra complexity that is not needed.

Challenges

Teams allow a team to manage the Product Backlog to a level that is more conducive to a team environment. The Team functionality takes advantage of the new Product Backlog and Storyboard functionality.

Development Teams that are using Area Path to define a component in their Team Project will not be prevented from doing so. They will need to understand the planning behind Area Paths starting with TFS 2012.

Team members cannot be assigned multiple roles in a Team within a sprint. For example, Abu who is the build master cannot be assigned to be a developer or a tester in addition to his build role within a sprint. This is a limitation of the Team functionality that might be addressed in a future release. A team member can switch roles between sprints however, not within a sprint.

Recommendations

Keep the teams manageable. Ideally, the team is working on specific functionality or is completely responsible for an application.

Options

Even Team Projects that don't fit the Scrum or Agile approach can benefit from having a Team defined for it to provide a quick glimpse into the work to be done, work in progress and work completed. Putting team members into a Team will get them accustomed to the Team approach and the benefits that will come with it.

Using the Trey Research Team Project as a single Team approach, Paul, Doris A, Doris B. and Christine can be assigned as members on the Trey Research Team. Mike, the Program Manager, will create the tasks in the Product Backlog and define the sprints. Preliminary assignments will be made by Mike with the Team members capacity kept in mind. In this scenario, the number of team members is a manageable number and having multiple teams would add extra complexity that is not needed.

Multiple Team Strategy

You can have a single application in a Team Project. The single application is supported by many developers who focus on different components of the application or on different release versions. When members of an application team have a specific focus it is easy to group these people into a team. These team members are not constrained to a single team. A database administrator is an example of a person who can be on several teams because they can have different roles on each team.

Example

Project Background

The Consolidated Messenger TFS Project is a web project supported by three development teams (Team A, Team B and Team C). The web project is a product sold to clients and is being re-designed from classic ASP to MVC. To keep all three teams involved with the transition to MVC while they continue to support the existing application, each team spends one month on production support tasks while the other two teams work on the new MVC functionality. Each month has two sprints for the new development work. The production support month is one sprint but there might not be a full month of work. If not, the team can work on new functionality for their upcoming sprints. Each team has a technical lead, a program manager, five developers, a tester, a build master and a database administrator. The technical lead does development tasks in addition to code reviews and the code reviews can be done for other teams, not just the team they are assigned to. The program manager tracks the team's progress, meets with the other project managers and the application owner representative to identify functionality of the new application as well as help with testing. The developers and database administrator are dedicated to the tasks that they select during their time in a sprint. However, they can change roles to be a tester in a different sprint. The build master has a development role during the first sprint of the month and is dedicated to builds and deployments during the second sprint of the month.

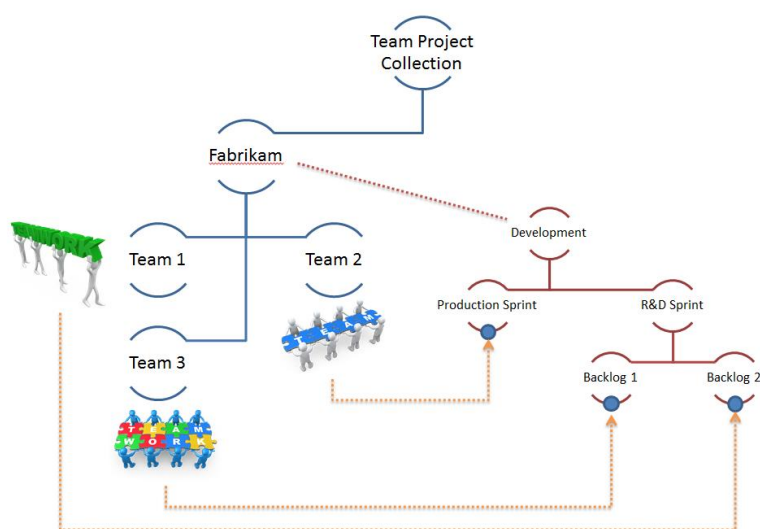


Figure 30 - Consolidated Messenger multiple team strategy showing an example month

Using the Consolidated Messenger Team Project as a multiple Team approach, each team has a sprint backlog they create from the product backlog. The team owns their backlog, which is defined by the area/iteration. During the month that a team is assigned to production support, they can pull items from a future sprint, as long as those items have been defined for them in that future sprint. For the teams working on new functionality, the Project Manager takes on testing responsibilities because the first sprint of the month focuses on development

and the second sprint of the month focuses on testing. At the start of the sprint, the Project Manager schedules a review with the team members and then adjusts each team member's capacities for that sprint.

Challenges

Deciding to take on work for a future sprint has inherent risk. What happens if you commit to the work and can't complete it? In the example of the production support team deciding to work on items from a future sprint, there's the chance that the production code will require changes and that production work will override any future work they committed to. Now they will have uncompleted future sprint work.

Just like with the single team approach, development teams that are using Area Path to define a component in their Team Project will not be prevented from doing so. They will need to understand the planning behind Area Paths, starting with TFS 2012.

Having a person on different teams can lead to capacity issues. The time commitment of one person across multiple teams can turn into a prioritization issue if it is not properly managed.

Recommendations

Keep the teams manageable. Ideally, the team is working on specific functionality or is completely responsible for an application.

Options

Even Team Projects that don't fit the Scrum or Agile approach can benefit from the Team approach. By Default there is always a team created for a Team Project. Using this Team will provide a quick glimpse into the work to be done, work in progress and work completed

References

- [Visual Studio ALM Rangers Quick Reference Guidance](http://vs2010quickref.codeplex.com/) ⁵⁸
- [Visual Studio ALM Blog: What's New for TFS11](http://blogs.msdn.com/b/visualstudioalm/archive/2011/09/20/visual-studio-team-foundation-server-11-developer-preview-what-s-new-for-team-foundation-server.aspx) ⁵⁹

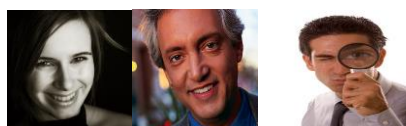
⁵⁸ <http://vs2010quickref.codeplex.com/>

⁵⁹ <http://blogs.msdn.com/b/visualstudioalm/archive/2011/09/20/visual-studio-team-foundation-server-11-developer-preview-what-s-new-for-team-foundation-server.aspx>

Defining your Disaster Recovery (DR) Strategy

This section introduces you to **disaster recovery (DR)** which will assist you in pro-active DR avoidance, planning, and recovery.

This section focuses on **Jane** the infrastructure specialist, **Dave** the TFS administrator, and **Paul** the database administrator.



I would like to ...	Page
• Understand DR Strategies	44
• Understand DR Avoidance	46
• Understand DR Recovery	51

Table 17 - Deciding what information is important with disaster recovery planning

Understand DR Strategies

What we understand with Disaster Recovery (DR)

Disaster recovery is the dreaded process of recovering infrastructures and services after disruption by a disaster, whether man-made⁶⁰ or natural⁶¹. Both are very difficult to predict and prevent, making it important for organizations to define, implement, and continuously evaluate processes, procedures, and policies to mitigate the risk to critical infrastructures and services.

Disaster recovery planning typically encompasses preventive, corrective, and detective measures. See:

- [TFS Installation and Administration Guide](#) ⁶²
- [Disaster Recovery Planning \(Database Engine\)](#) ⁶³
- [Be Prepared: A Guide to SharePoint Disaster Prevention and Recovery](#) ⁶⁴
- [Plan for disaster recovery \(SharePoint Server 2010\)](#) ⁶⁵

DR Avoidance Strategy ... seeing the smoke before the fire

Although this guide mentions the possible DR Strategies for TFS, it focuses on detective measures to prevent an avoidable disaster recovery caused by infrastructure and solution degradation. It's imperative that you seriously consider disaster recovery strategies, because even the most pro-active and best detective measures cannot guard you against man-made or natural disasters.

⁶⁰ Man-made disaster examples: war, terrorism, hacking and negligence

⁶¹ Natural disaster examples: earthquake, flood and meteor impact

⁶² <http://www.microsoft.com/en-us/download/details.aspx?id=29035>

⁶³ [http://technet.microsoft.com/en-us/library/ms178128\(v=sql.100\).aspx](http://technet.microsoft.com/en-us/library/ms178128(v=sql.100).aspx)

⁶⁴ <http://technet.microsoft.com/en-us/magazine/2005.11.beprepared.aspx>

⁶⁵ <http://technet.microsoft.com/en-us/library/ff628971.aspx>

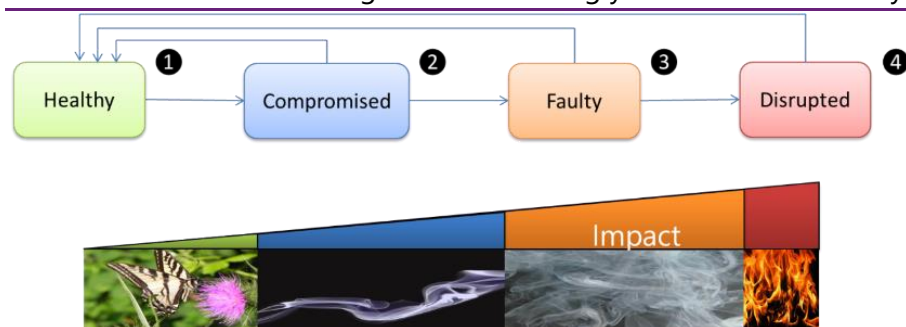


Figure 31 – Impact of system failure from smoke to fire

As shown in Figure 31, a system can go through four typical states of health. We start with a **1** healthy system in which all services and hardware function correctly, switch to **2** a compromised system when software or hardware encounters issues that are not evident to the user, degrade to **3** a faulty system when users notice errors, and eventually are disrupted by **4** by a disastrous failure.

It's our objective to help you detect the "puff" of smoke before you are faced with thick plumes of smoke and, eventually, a raging forest fire that is complex and costly to manage. By pro-actively detecting and managing issues in your TFS you are minimizing the impact of cost and downtime within your operations, resulting in productive and happy TFS users.

DR Strategies

The following table summarizes the disaster recovery (DR) strategies that can be considered for TFS and the recommended⁶⁶ consideration for use in the Humongous Insurance (HI), Consolidated Messenger (CM) and Trey Research (TR) customer types.

Strategy	Description	HI	CM	TR
Simple	In an environment where you are prototyping or are not dependent on TFS maintaining history, you can adopt a simple DR strategy that is based on re-installing TFS and checking in the latest known codebase. This codebase is typically backed up on multiple systems, removable disk drives and/or backups.			✓
Backup and Restore	Refer to the TFS Administration Guide ⁶⁷ , section Backing up and Restoring Your Deployment , to determine the best backup strategy. TFS 2012: Refer to the Microsoft Visual Studio Team Foundation Server 2012 Power Tools ⁶⁸ which contain the TFS Backup/Restore tool. This tool can schedule backups and use the restore wizard to perform a selective or complete restore. TFS 2012 Update 2: The Backup/Restore PowerTool functionality is now in the product and referred to as Scheduled Backups. This feature can be accessed from the TFS Administration Console.		✓	✓
AT Standby Servers	To avoid downtime in the event of an Application tier (AT) server failure or AT server maintenance you can implement a Standby Application tier. Refer to TFS Installation Guide ⁶⁹ , section Configuring a Standby Application tier for more information.	✓	✓	

⁶⁶ All of the DR strategies are feasible for all customer types. The recommended consideration is based on practicality, considering complexity and cost of infrastructure and support.

⁶⁷ <http://www.microsoft.com/en-us/download/details.aspx?id=29035>

⁶⁸ <http://visualstudiogallery.msdn.microsoft.com/b1ef7eb2-e084-4cb8-9bc7-06c3bad9148f>

⁶⁹ <http://aka.ms/tfsinstallguide>

Planning Guide – Defining your Disaster Recovery (DR) Strategy

Strategy	Description	HI	CM	TR
Team Foundation Cluster	To avoid downtime in an event of a Data tier (DT) server failure or DT server maintenance you can implement a Team Foundation Cluster. Refer to TFS Installation Guide , section Configuring Servers for Team Foundation Cluster Installation for more information.	✓	✓	

Table 18 – DR strategies

NOTE

Backup/Restore PowerTool is not supported on releases where Scheduled Backups is available? It's either you use one or the other. If you upgrade to a build that has Scheduled Backups from one that uses the PowerTool, your PowerTool settings will be automatically configured into Scheduled Backups and the PowerTool uninstalled as part of the upgrade.

Determining your DR Strategy

To determine if you need Simple, Backup and Restore, AT Standby Servers and/or Team Foundation Cluster services, use the following basic questions.

Question	Simple	Backup	Standby	Cluster
Is TFS history information not crucial?	✓			
Do you rely on TFS history information?		✓	✓	✓
Do you rely on avoiding Application tier (AT) downtime during maintenance or AT server failures?			✓	
Do you rely on avoiding Data tier (DT) downtime during maintenance or DT server failures?				✓
Do you have an objective for 99.999% availability of your TFS system?		✓	✓	✓

Table 19 – Determining your DR strategy

Refer to **Planning the ideal DR environment for TFS**, on page 80, for a more detailed planning walk-through.

Understand DR Avoidance

DR Avoidance is about monitoring your system, noticing and dealing with “smoke” before it turns into a “raging fire,” commonly referred to as a disaster.

Tools that help you troubleshoot and monitor

Tool	Description	Reference
Microsoft Baseline Security Analyzer	This tool will help you identify missing security updates and common security misconfigurations in Windows, SQL Server, and IIS. Used regularly, this tool helps increase system reliability.	<ul style="list-style-type: none"> How To: Use the Microsoft Baseline Security Analyzer ⁷⁰ Microsoft Baseline Security Analyzer 2.2 (for IT Professionals) ⁷¹

⁷⁰ <http://msdn.microsoft.com/en-us/library/ff647642.aspx>

⁷¹ <http://www.microsoft.com/en-us/download/details.aspx?id=7558>

Planning Guide – Defining your Disaster Recovery (DR) Strategy

Tool	Description	Reference
Microsoft TFS 2010 Best Practices Analyzer	This tool collects TFS data from your environment and produces a comprehensive 'best practice' rules report. Use this tool before installing or upgrading TFS, thereafter on a regular basis, and while troubleshooting.	<ul style="list-style-type: none"> • Best Practices Analyzer Tool for Team Foundation Server ⁷² • Team Foundation Server Power Tools December 2011 (TFS BPA download) ⁷³
SCOM TFS Monitoring Management Pack	<p>Proactive and reactive monitoring of TFS can be done by using the Microsoft System Center Operations Manager (SCOM) together with a Management Pack designed for TFS.</p> <p>Feature Summary (from download link below)</p> <p>The monitoring provided by this management pack includes availability and configuration monitoring, performance data collection, and default thresholds. You can integrate the monitoring of TFS components into your service-oriented monitoring scenarios.</p> <ul style="list-style-type: none"> • Auto discovery of TFS components. • Implements a containment hierarchy, reflecting logical architecture of the Product. • Implements a proper health model using Monitors. • Contains tasks, diagnostic and recovery for certain failures. • Provides events that indicate service outages. • Provides alerts that show configuration issues and connected data source changes. • Verification that all dependent services are running. • Triggers targeted running of BPA against TFS Servers from Operator Console in TFS 2008 and 2010. 	<ul style="list-style-type: none"> • TFS2012: Visual Studio 2012 Team Foundation Server Monitoring Management Pack ⁷⁴ • TFS2010: Visual Studio 2010 Team Foundation Server Monitoring Management Pack ⁷⁵ • TFS2008: Visual Studio Team System 2008 Team Foundation Server Management Pack for System Center Operations Management 2007 ⁷⁶
Performance Analysis of Logs (PAL) Tool	Reads a performance monitor counter log and analyzes it using known thresholds. It's a great tool to use when you need to investigate (potential) performance issues in your environment but are not familiar with the various performance counters available.	<ul style="list-style-type: none"> • Performance Analysis of Logs (PAL) Tool ⁷⁷
Reports	Reports (based on SQL Server Reporting Services) that can be used to evaluate and get a picture of the status of some of the internals of your TFS environment. Download the administration reports and upload them to your TFS Reporting Services environment by following the instructions found in the links.	<ul style="list-style-type: none"> • Report pack: TFS2010: Warehouse and Job Service Administrator Reports ⁷⁸ • Report pack: Administrative Report Pack for Team Foundation Server 2010 ⁷⁹ • Blog: Monitoring the TFS Data Warehouse - FAQ ⁸⁰ • Blog: Data Driven Subscription Reporting a la Grant ⁸¹

⁷² [http://msdn.microsoft.com/en-us/library/ee248630\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/ee248630(v=vs.100).aspx)

⁷³ <http://visualstudiogallery.msdn.microsoft.com/c255a1e4-04ba-4f68-8f4e-cd473d6b971f>

⁷⁴ <http://www.microsoft.com/en-us/download/details.aspx?id=35773>

⁷⁵ <http://www.microsoft.com/en-us/download/details.aspx?id=6325>

⁷⁶ <http://www.microsoft.com/en-us/download/details.aspx?id=14720>

⁷⁷ <http://pal.codeplex.com>

⁷⁸ <http://blogs.msdn.com/b/granth/archive/2010/02/07/tfs2010-warehouse-and-job-status-reports.aspx>

⁷⁹ <http://blogs.msdn.com/b/granth/archive/2010/07/12/administrative-report-pack-for-team-foundation-server-2010.aspx>

⁸⁰ <http://blogs.msdn.com/b/granth/archive/2010/07/12/monitoring-the-tfs-data-warehouse-faq.aspx>

⁸¹ http://blogs.msdn.com/b/willy-peter_schaub/archive/2011/01/31/tfs-integration-platform-data-driven-subscription-reporting-a-la-grant.aspx

Table 20 - Tools that help you troubleshoot and monitor

Performance Counters worth monitoring

The section **Working with Team Foundation Server Performance Counters** on page 77 has a complete list of performance counters that can determine the health of your ecosystem. The following performance counters and the specified thresholds should be monitored as part of your DR avoidance strategy.

Processor utilization

Counter	Threshold
% Processor Time	Should be less than 80% (Minor peaks over 80% are OK)
% Privileged Time	Should be less than 25% of total processor time

Table 21 – Processor utilization performance counters

Memory utilization

Counter	Threshold
Available MBytes	Should be greater than 10% of total RAM
Pages/sec	Should be less than 2,500 pages per second

Table 22 – Memory utilization performance counters

Disk

Counter	Threshold
Avg. Disk Read/sec	Should be less than 10-25 ms
Avg. Disk Write/sec	Should be less than 10-25 ms
Logical disk/Free megabytes	System Partition greater than 500 MB (10%). Non system Partition greater than 2 000 MB (10%) ⁸²

Table 23 – Disk performance counters

Network

Counter	Threshold
Bytes Total/sec	Network Utilization should be less than 40% of the total bandwidth and anything above 65% is critical. This is how you calculate % Network Utilization: ⁸³ $((\text{Bytes Total /Sec} * 8) / \text{CurrentBandwidth}) * 100$
Packets Outbound Errors	Should be 0
Output Queue Length	Should be less than 1. A value greater than or equal to 1 is a sign of packets queuing on the NIC.

Table 24 – Network performance counters
⁸² <http://technet.microsoft.com/en-us/library/dd262028.aspx>
⁸³ <http://blogs.technet.com/b/kevinholman/archive/2011/03/02/how-to-collect-performance-data-from-a-script-example-network-adapter-utilization.aspx>

Web Performance

Counter	Threshold
ASP.NET Applications(*)\Requests In Application Queue	Should be as low as possible aiming for 0
ASP.NET Applications(*)\Request Execution Time	Benchmark your environment when performance is good, to determine the ideal threshold (as low as possible) for your environment.
TFS Services\Average Response Time	
TFS Version Control\Average Response Time	

Table 25 – Web performance counters

System

Counter	Threshold
Context Switches/sec	Should be less than 5000 per processor and more than 10000/processor indicates a constraint
Paging File(*)\% Usage	Should be less than 70%
Total Cache Hits	Benchmark your environment when performance is good, to determine the ideal threshold (as high as possible) for your environment.

Table 26 – System performance counters

SQL server

Counter	Threshold
Buffer Manager\Page reads/sec	Should be less than 90

Table 27 – SQL Server performance counters

Reports for Monitoring TFS health

NOTE

Refer to **Authoring Reports**, page 97 for example reporting walkthroughs

Report	Description	Purpose / Reference Link
Average Build Duration	This report provides details of the average time taken by successful builds. This will help in identifying and monitoring the build information in tabular format.	Average duration for successful builds per build definition.
Blocked Field Changes	This report shows blocked fields that have conflicts over all Team Project Collections.	Conflicts across all TPCs for fields being blocked. Administrative Report Pack for Team Foundation Server 2010 ⁸⁴
Build agent hourly distribution	This report provides the builds run per hour for a particular TFS Project.	Hourly distribution of the usage of the build agents.

⁸⁴ <http://blogs.msdn.com/b/granth/archive/2010/07/12/administrative-report-pack-for-team-foundation-server-2010.aspx>

Planning Guide – Defining your Disaster Recovery (DR) Strategy

Report	Description	Purpose / Reference Link
Build server summary	This report will provide the build server usage details and details on the number of builds run and their durations.	Statistics on a build server.
Cube status	This report provides the following information: <ul style="list-style-type: none"> How long is cube processing taking? How much time elapses between processing jobs? How often do the processing jobs run? Do errors occur when the cube is processed? 	Used to monitor Analysis Cube processing that occurs on a regular schedule. Administrative Report Pack for Team Foundation Server 2010 ⁸⁵ .
Execution time for user	This report provides a visualization of the load by total execution time on the server from and provides details on the users who are putting the biggest load on the server.	Breakdown of user execution time.
Execution time summary	This report provides a visualization of the load by total execution time on the server from two axes: users and commands. You can use this report when you want to know: <ul style="list-style-type: none"> Which commands account for the largest load on the server? Which tools or users are putting the biggest load on the server? 	Breakdown of execution time.
Job status	The Job status report shows the job definitions for the instance and the interval they're set to run on. This is useful for checking to see if a job has somehow been disabled or changed. The report also shows the Job History.	Job definitions for the instance and the interval they're set to run on from Warehouse and Job Service Administrator Reports ⁸⁶ .
Reportable fields / Queued fields changes	The Reportable Fields report shows all reportable fields in the deployment of TFS. Administrators of Team Projects can use this report before they add a reportable field or change the properties of an existing field to prevent potential schema-merge conflicts. It lists fields across all collections, including any fields that are blocked. The Queued Filed Changes report shows field changes that are queued behind the blocked changes.	Administrative Report Pack for Team Foundation Server 2010 ⁸⁷ .
Server status - historical performance trends	This report serves as a summary of the average response time for two of the TFS subsystems: Work Item Tracking and Version Control. You can use this report when you want to know: <ul style="list-style-type: none"> How long are users, on average, waiting for a subsystem to process their request. Which days of the week are the most critical when it comes to performance. 	Performance monitoring.
Server status - recent performance trends	This report provides more granularity about the performance of the server. The reports start with a view of the server average response time, looking at the entire picture instead of response time broken down by subsystem. This is followed by charts about version control downloads and average response time distributions for the same time period. Use this report when you want to know: <ul style="list-style-type: none"> The correlation between degraded server performance and average response times by subsystem. 	Performance monitoring.

⁸⁵ <http://blogs.msdn.com/b/granth/archive/2010/07/12/administrative-report-pack-for-team-foundation-server-2010.aspx>

⁸⁶ <http://blogs.msdn.com/b/granth/archive/2010/02/07/tfs2010-warehouse-and-job-status-reports.aspx>

⁸⁷ <http://blogs.msdn.com/b/granth/archive/2010/07/12/administrative-report-pack-for-team-foundation-server-2010.aspx>

Planning Guide – Defining your Disaster Recovery (DR) Strategy

Report	Description	Purpose / Reference Link
	<ul style="list-style-type: none"> How a large number of downloads affects overall server performance. An overall health indicator of the server. 	
Server status - Source control request queue	<p>This report provides information about:</p> <ul style="list-style-type: none"> Whether a request blocked source control operations and for how long. How healthy the performance of version control on this hardware is. 	Performance monitoring.
Server Status - Top users bypassing proxies	This report allows administrators a view into which users are not complying with internal guidelines around proxy usage, which decreases overall server performance.	Users not using the proxy server.
TFS usage	This report provides information on the total number of users using the farm. This will help monitor the load on server.	Number of users using a TPC.
Warehouse status	The warehouse report provides a quick and easy way to find out if an incremental or full analysis processing is in progress. It also shows any errors (like warehouse schema conflicts) in the 'Last Run' column. This report is also useful after an upgrade or when the warehouse needs to be rebuilt manually. It shows each of the data adapter sync jobs for each collection and their current status. During normal operation, these will run very quickly as data changes in the operational stores, and probably always appear "Idle." It will also show any errors from previous job executions in the 'Last Run' column.	Warehouse and Job Service Administrator Reports ⁸⁸ .
Average response time	This report provides the average response for requests made by users. This report will help in understanding the performance of the farm. As a standard, a lower average response time signifies a good health of the farm.	Performance monitoring.
SQL connection failures/sec	This report provides the details of the connectivity issues between the Application and Data tiers. As a standard, a lower number of failures signifies good farm health.	Report to Check the SQL issues ⁸⁹
CPU utilization	This report provides the details of the CPU utilization of the Application tier. As a standard, a lower number of CPU utilization signifies that the system is not under load and can support additional users.	Monitor CPU Utilization on App and Data tier
Available memory	This report provides details of the RAM utilization of the application and Data tier. This report can be used to monitor the free RAM in the servers, which is important for smooth operation of the farm.	SQL and TFS App tier memory usage
Requests/second	This report provides details on the user load on the system. A higher RPS signifies a higher load on the system.	Performance monitoring.

Table 1 - Reports for Monitoring TFS health

DR Recovery Walkthroughs

If you are in the unfortunate position of being confronted with a fire (disaster), we have created walk-throughs for you to step through the recovery process.

⁸⁸ <http://blogs.msdn.com/b/granth/archive/2010/02/07/tfs2010-warehouse-and-job-status-reports.aspx>

⁸⁹ <http://blogs.msdn.com/b/granth/archive/2008/11/07/querying-perfmon-data-from-sql.aspx>

Planning Guide – Defining your Disaster Recovery (DR) Strategy

Disaster Recovery Scenario	See page
AT Failure	87
Build Failure	92
Complete Failure (aka Datacenter fire or meteor strike)	84
Failover to a secondary site	87
SQL Server Dies (DT Failure)	85
Proxy Failure	91

Table 28 - Disaster recovery walkthroughs

Large and Complex Project considerations

This section describes additional learning from in-the-field and considerations for large and complex project environments, which includes differentiations from other environments such as:

- Larger solutions, infrastructures and teams working on one or more projects.
- More complexity in terms of business processes and associated project environments.
- More dependencies between customer types, such as Humongous Insurance, Trey Research, and Consolidated Messenger.
- More team environments and variations between environments.

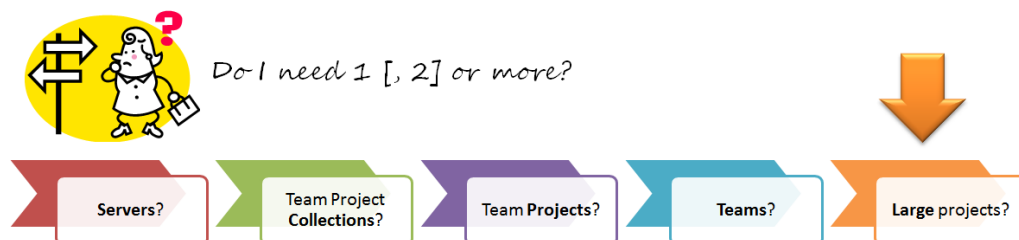


Figure 32 - Large projects planning

This section focuses on **Dave** the TFS administrator, **Jane** the infrastructure specialist, and **Garry** the lead for development teams.



I would like to ...	Page(s)
• Understand Large Projects	54
• Key Decisions: When to use what	56
• Explore a hypothetical organization with a large project	57
• Explore a hypothetical organization with Legacy Systems	59

Table 29 - Deciding what information is important with large and complex team planning

Understanding Large Projects

What makes a project large?

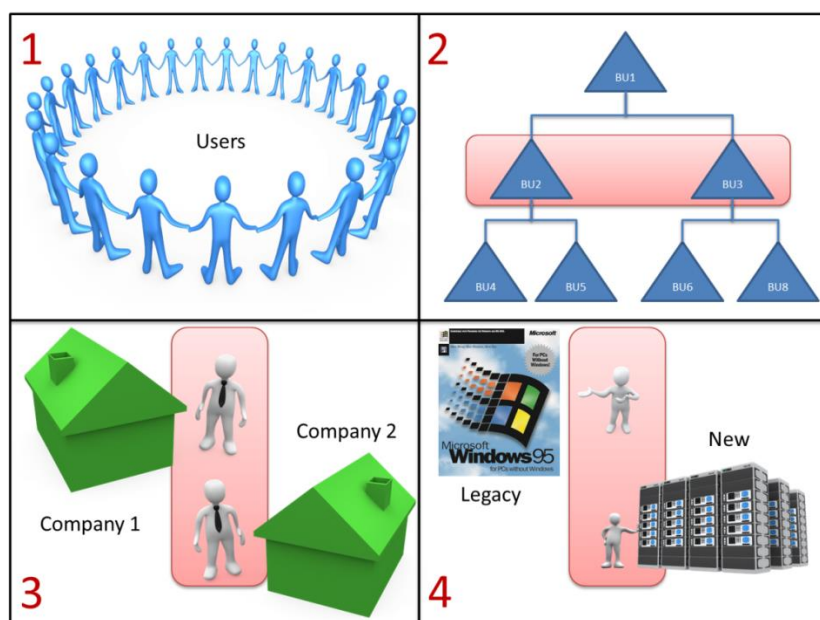


Figure 33 – What makes a project large?

The definition of a large and potentially complex project varies from organization to organization. **Figure 33** shows four typical large project definitions we have encountered:

1. Projects with a **large number** of concurrent **users** (100+), which result in spikes of builds/day and place pressure on services, such as Reporting Services.
2. Projects that involve a number of **business units** with separate responsibility boundaries and **cross-business unit** development within a single organization.



Humongous Insurance customer type would be a good example.

3. Projects that involve a number of **business units** with separate responsibility boundaries and **cross-business unit** development, including a cross-company relationship. Examples would include software development company X, consultancy company Y and in-house development teams from the customer.



Consolidated Messenger customer type would be a good example.

4. Projects with a combination of new and **legacy systems** which are maintained or migrated to the new system.

This section walks through a hypothetical example of the last three definitions.

NOTE

The hypothetical examples are intended to be used as a starting point when you explore potential solutions, not as best practice templates.

What are the challenges of large projects?

As projects grow in size and complexity, they introduce additional challenges such as:

- **Builds** become more numerous, complex and costly to administrate. As a project gets larger and especially as cross-team development scenarios emerge, administrators have to deal with:
 - Dependencies between Team Project Collections, Branches and Build servers
 - Increased builds and long-running builds, which can introduce unproductive queuing and loading of the build servers.

See [Visual Studio ALM Rangers Build Customization Guidance](#) ⁹⁰ for more information.

- **Branching and Merging** becomes more complex, introducing time and resource consuming forward (main→dev) and reverse (dev→main) integration, which usually emerges as build failures, missed milestones and features that fail to propagate to the Main branch.

See [Visual Studio ALM Rangers Branching and Merging Guidance](#) ⁹¹ for more information.

- **Teams and dependencies** between boundaries of responsibility become more difficult to manage.
 - The size and number of teams typically start to grow with larger projects and introduce collaboration and dependencies, which must be carefully considered to avoid complexity and competition for resources.
 - Distributed and virtual team ecosystems emerge as team boundaries extend beyond on-premises environments, which introduce unique infrastructure, management and collaboration challenges.
 - The choice of Infrastructure, Team Project Collection, Team Project and Team strategies start to play a pivotal role. Ensure that you plan early and observe and evolve continuously.

See [Visual Studio ALM Rangers - Reflections on Virtual Teams](#) ⁹² to get a view of managing virtual teams.

Deciding on a Large Project Strategy

By reviewing the Team Project Collection, we can find some facts that will assist us in our decisions about how to structure our large and complex organizations.

Different Isolation Levels

When we talk about isolation levels there is a clear difference between a Multiple and a Single Collection strategy:

Different Isolation levels	
Multiple Collections	Single Collection
Isolation is total between Collections. Process Templates, Reports, Work Items, Queries, Version Control, Build Services are inaccessible over the collection boundary.	Isolation is configurable at Team Project level. Almost everything is possible to share, including Process Templates, Reports, Work Items, Queries, Version Control and Build Services.
Separate security setting for each Collection that defines the top level security for each contained Team Project.	Top level security is set on Collection and propagated to the Team Projects. Each Team Project could define modifications of / additions to the top level security.

Table 30 – Clear isolation level difference between multi and single collection strategy

⁹⁰ <http://go.microsoft.com/fwlink/?LinkID=230938>

⁹¹ <http://go.microsoft.com/fwlink/?LinkID=230936>

⁹² http://blogs.msdn.com/b/willy-peter_schaub/archive/2011/09/07/visual-studio-alm-rangers-reflections-on-virtual-teams.aspx

Single versus Multiple Team Project Collection Strategies

The advantages of using Multiple or Single Collections	
Multiple Collections	Single Collection
Possible to implement a more complex isolation strategy.	Cross Team Project sharing or linking of Work Items, Queries and Reports are possible.
Starting, stopping, backups are possible on individual Collections. It affects only Team Projects in that Collection.	Cross Team Project sharing of content in Version Control is possible (Move/Branch/Merge).
Possible to distribute Collection databases across SQL Server instances.	Cross Team Project working is possible.
	Single database for all Team Projects data. Easy to maintain.

Table 31 - The advantages of using multiple or single collections

Looking from the opposite view, we can of course also see some disadvantages:

The disadvantages of using Multiple or Single Collections	
Multiple Collections	Single Collection
No cross Collection sharing of Work Items, Queries and Reports.	Single database for all Team Projects data.
No cross Collection Version Control (Move/Branch/Merge).	Restore operations of database affects all Team Projects.
No cross Team Project work is possible if Team Projects are located in different Collections.	Each Collection has its limited capacity of Team Projects.
Cannot share Build Services between Collections.	
30-100 active Collections per SQL Server instance limitation.	

Table 32 – The disadvantages of using multiple or single collections

Key Decisions - When to use what

Based on the guidance in sections Deciding on a Team Project Strategy, on page 26 and the discussions of Single versus Multiple Team Project Collection Strategies on page 56, we have created a suggested “when to use what” list for large and complex project environments:

When to use Multiple or Single Collections	
Consider Multiple Collections when ...	Consider Single Collection when ...
The need for isolation is preferred	The need for collaboration is preferred
There are many Team Projects	There are few Team Project
There are many users or large teams	There are few users or small teams
You have a global organization	You have a localized organization
You have distributed teams	You have local teams
You have a complex organization of projects in projects etc.	You have a simple organization

Table 33 – When to use multiple or single collections

As we have already seen there is no clear definition of when to use a specific approach.

What we can say with certainty is that we typically have some high level grouping where, in most cases, some isolation is required. For these groups we recommend using separate Collections.

In the mid-level, we typically find items that in some way are related to each other. This could be Clients, Releases, Products or Business Units. At this level we recommend using separate Team Projects.

At the lowest level, we typically find some logical units that, combined, add up into one of the mid-level items. This is the level where we recommend using Components (or Area if backward compatibility is used).

Team Collection	Team Project	System Component
City	Client	Project
Product	Release	Components (Area)
Product	Business Unit	Components (Area)
Business Unit	Product	Components (Area)
Customer	Business Unit	Project

Table 34 – Grouping differences

Strategy 1 - Single Team Project Collection examples

Humungous Insurance example:

A global company with distributed teams working on multiple products that are used and occasionally shared within several business units of the company. The actual work is performed by people from a multitude of countries. The members are jumping back and forth between different projects and maintenance for the products. They could even be members of several teams at the same time. To ensure reusability of source code between the different products, collaboration and standardization are considered to be of the highest importance. For this purpose, the Single Team Project Collection strategy is a good option.

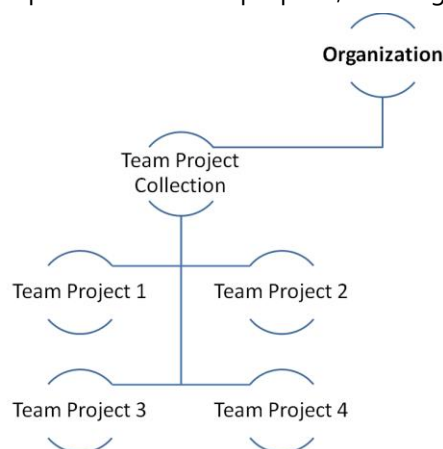


Figure 34 – Large project: single Team Project Collection strategy

Each product is contained within its own master Team Project.

Separate Team Projects are used for maintenance. Each project contains branches from a collection of different products' master Team Projects. Each team keeps its own backlogs and reports. The maintenance for all products they are responsible for is grouped together.

Separate Team Projects are also used for each project. Team Projects contain branches from a collection of different products' master Team Projects that are included in that specific project. Each Team Project also keeps its own backlogs and reports grouping the project work together.

Each master Team Project is connected to the different projects and maintenance queries and reports. Having a master Team Project enables follow up on the lifecycle of the product.

The administration of members and their team assignments is easily maintained on collection level and / or Team Project level. The members could be allowed read access to the whole collection and at the same time have extended rights to each specific Team Project that they are specifically involved in.

To increase availability, a centralized instance of TFS is used in combination with TFS Proxies on locations where bandwidth problems exist.

The build environment consists of centralized Build Controller / Agents with the drop shares placed at local locations. This ensures a standardized way of building as fast as possible at the same time as the build output is delivered to local locations for testing and deploying.

Strategy 2 - Multiple Team Project Collections examples

Consolidated Messenger

A consultant company is working within several cities with locally located consultants and clients. A decision is made to isolate the different cities. This is necessary to ensure that the availability for each city isn't affected by the others. Consultants working in the different cities should not have access to the other cities' client material.

However, within each city, the consultant company could have one or several clients. In this situation, sharing artifacts among the clients would be useful in terms of common components, build environments etc.

For this purpose, the Multiple Team Project Collection is a good option.

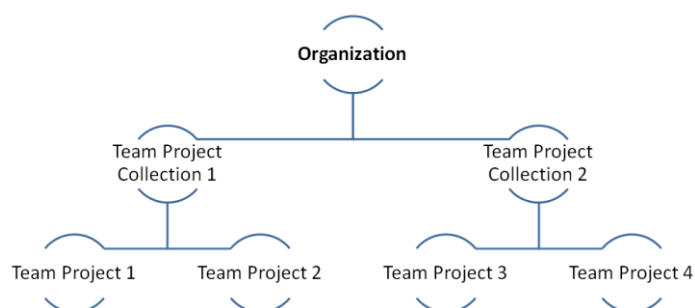


Figure 35 – Large project: multiple Team Project Collection strategy

To keep a high level of isolation, each city has its own collection. By using Multiple Team Project Collections, it is possible to back up and restore each city's database without affecting the other cities.

It's also possible to ensure that only consultants in a specific city are given access to that city's different clients. The other cities and clients are totally invisible from the outside.

Each client is kept in a separate Team Project with Areas defined for the different projects and maintenance.

It's easy for a single consultant to participate in several teams, working for several clients concurrently.

It's easy to share different artifacts between the clients that the consultants are working with.

Each city is responsible for its own build environment, making it possible to have a custom made build process that covers their own client's specific needs.

Migrating and Coexisting with Legacy Systems

Basic Decisions

We recommend dealing with the “legacy” scenario that is based on the Team Project guidance; see page 26 for details. In essence:

- If the legacy code requires different **security models**, different **process models** or different **ownership**, we consider a separate Team project.
- If we split based on ownership:
 - If the **same team** is working the legacy code, consider one Team Project for all legacy code.
 - If **different teams** are working on different parts of the legacy code, for example different legacy solutions, consider one Team Project for all legacy code and using Teams to segregate teams.
 - If **different teams** from **different organizations** are working on different parts of the legacy code, for example different legacy solutions, consider a Team Project per organizational business unit.

Examples

Single Team Project Collection strategy and Legacy Code

Single Team Project strategy – Trey Research

As outlined on page 27 and in particular page 34, **Trey Research** might decide on a Single Team Project strategy, if all the specialized solutions for numerous clients are based on the same code, with only slight modifications to the base solution.

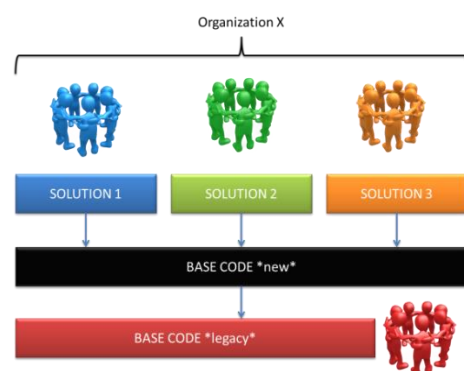


Figure 36 - Single new and legacy code base

When legacy code is included in the solution, we recommend that it is maintained by a separate team in its own Team Project. This ensures that the legacy code will **not** remain entwined in the solution code base once it is retired, because it is not a trivial task to split a Team Project and its artifacts into multiple Team Projects.

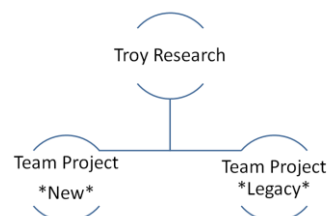


Figure 37 - Single TPC & TP, with legacy code - Trey Research

Multiple Team Project strategy – Humongous Insurance

As outlined on page 27 and in particular page 34, **Humongous Insurance** might decide for the Multiple Team Project strategy, if they are developing several independent solutions.

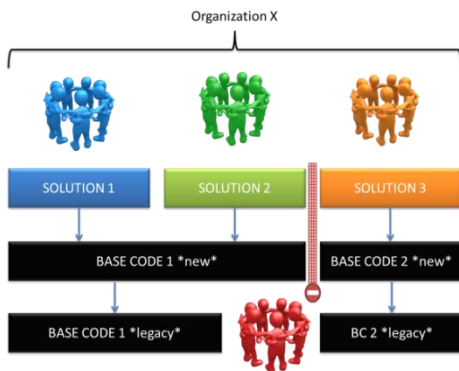


Figure 38 - Single new and legacy code base

As with the previous example, we recommend that legacy code is maintained by a separate team in its own Team Project. This ensures that the legacy code will **not** remain entwined in the solution code base once it is retired, because it is not a trivial task to split a Team Project and its artifacts into multiple Team Projects. In the case of Humongous Insurance we have, as shown in Figure 38, two legacy and two new solution code bases, which results in the following possible Team Project design:

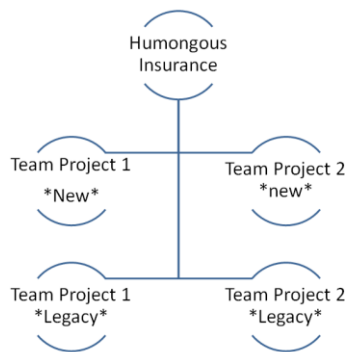


Figure 39 - Single TPC & Multi-TP, with legacy code – Humongous Insurance: example 1

If, for example, we had one new solution, based on three separate legacy code bases, the Team Project design would change as follows:

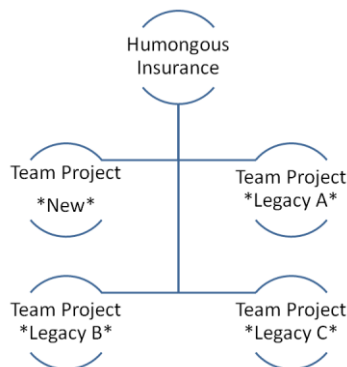


Figure 40 - Single TPC & Multi-TP, with legacy code – Humongous Insurance: example 2

Multiple Team Project strategy – Consolidated Messenger

As outlined on page 27 and in particular page 34, **Consolidated Messenger** might decide for the Multiple Team Project and Team Project Collection strategy if they have several development groups developing solutions for several customers. This is primarily to create an isolation, management and ownership barrier between organizations.

Figure 41 is just one of many possible Team Project designs, which creates a clear **isolation** barrier between the in-house and external projects using separate Team Project Collections and **categorization** of projects within each Team Project Collection.

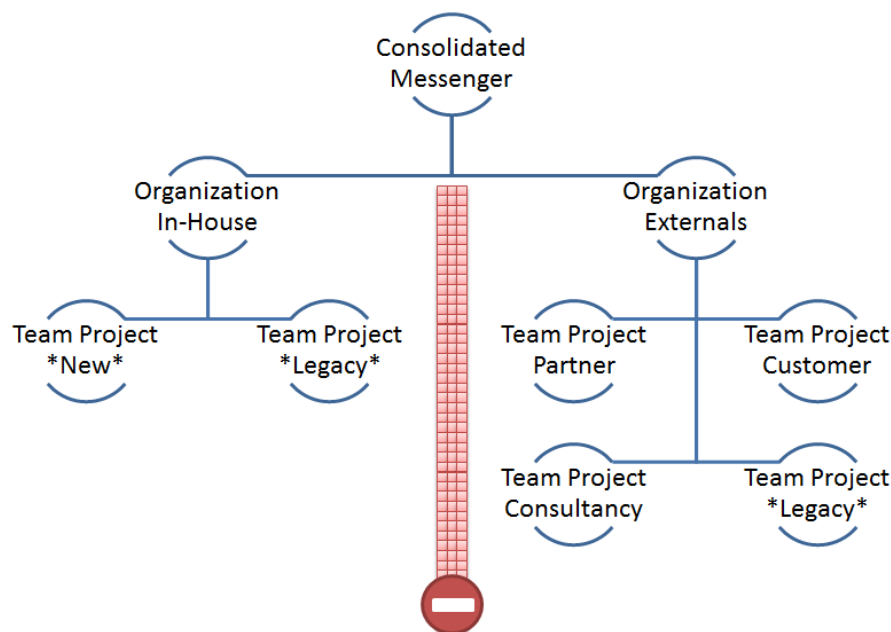


Figure 41 - Multiple TPC & TP, with legacy code – Consolidated Messenger

Key observations from the above diagram:

- Organizations are isolated from each other through the use of Team Project Collections.
- Groups, such as partner, customer and consultants, which potentially have different security models or process templates are contained in separate Team Projects.
- Legacy project is contained in its own Team Project, to split *new* from *legacy* code bases.

References

- [Visual Studio ALM Rangers Quick Reference Guidance](http://vs2010quickref.codeplex.com/) ⁹³
- [Visual Studio ALM Rangers Branching and Merging Guidance](http://go.microsoft.com/fwlink/?LinkID=230936) ⁹⁴
- [Visual Studio ALM Rangers Build Customization Guidance](http://go.microsoft.com/fwlink/?LinkID=230938) ⁹⁵
- [Visual Studio ALM Rangers – Reflections on Virtual Teams](http://blogs.msdn.com/b/willy-peter_schaub/archive/2011/09/07/visual-studio-alm-rangers-reflections-on-virtual-teams.aspx) ⁹⁶
- Agile Software Engineering with Visual Studio, Addison-Wesley Publisher
ISBN 978-0-321-68587-8

⁹³ <http://vs2010quickref.codeplex.com/>

⁹⁴ <http://go.microsoft.com/fwlink/?LinkID=230936>

⁹⁵ <http://go.microsoft.com/fwlink/?LinkID=230938>

⁹⁶ http://blogs.msdn.com/b/willy-peter_schaub/archive/2011/09/07/visual-studio-alm-rangers-reflections-on-virtual-teams.aspx

Questions & Answers

Capacity Planning

What if we have more than 20K-30K developers working on projects? How can we capacity plan for such deployments?

To accommodate this many users, you'll have to create multiple instances of TFS or consider a scale-out model with, for example, 20 Team Project Collections with 1000 users each. Start by categorizing the related projects and determine if you need to plan the environment based on active Team Project Collections, Team Projects, users or a combination thereof.

Security Isolation

What are the possible options to restrict access to the SQL Analysis Services Cube that TFS utilizes in a multi Team Project Collection scenario?

It depends on what is meant by restricting access. If the users are not members of the TFSWarehouseDataReader role in the Tfs_Analysis database they will not be able to query the cube directly (well, unless they are admins). For Reporting Services, you would need to remove— or not grant— them permissions in the project collection folders on the reporting website that you don't want them to see.⁹⁷

[Grant Team Members Access to the Analysis Services Cube](#)⁹⁸

If the question is "How do I prevent users from collection A from seeing warehouse data from collection B?" we do not believe there is a way to apply permissions at the collection level in the cube. Permissions are all or nothing.

How do I "fix" this?

When scaling out, SQL Report Server complained about an unsupported feature.

Context: We are running a TFS instance as a Single-Tier and decided to scale out to a Multi-Tier solution because of our increasing amount of users and Team Projects. When scaling out our TFS instance, we ran into trouble with the SQL Report Server saying it was an unsupported feature. How could this happen? We are following the TFS Guides?

Answer: One of the possible reasons is that you are running SQL Report Server on a SQL Server Standard Edition. By upgrading it to an Enterprise Edition this problem could be solved. The operation could be done without disturbing the existing instance simply by running the Enterprise Edition installation media and following the upgrade wizard.

Ref: <http://www.sqldev.org/sql-server-reporting-services/how-to-upgrade-sql-server-reporting-services-2008-standard-edition-to-enterprise-edition-on-the-fron-13865.shtml>

When renaming a folder which is included in a Continuous Integration build definition, builds may fail

Context: This question is not really a question but more of informational type. We had a Team Project where they renamed a folder included in the workspace for a Continuous Integration Build Definition. A funny thing

⁹⁷ ALMTalk, George Archer, Senior Support Escalation Engineer, Microsoft

⁹⁸ <http://msdn.microsoft.com/en-us/library/ms244642.aspx>

happened because all of a sudden the TFS lost control of that build's workspace and reverted back to the version control root so every time someone checked in something in another Team Project; it triggered the erroneous build which of course failed because the folder defined in the workspace didn't exist anymore. As a result, we ended up with approximately 6000 failing builds triggered during a 24 hour period.

Answer: The solution we decided to try was to create a PowerShell-script that runs as a task to make a sanity check of all the build definitions. We wanted to make sure that the workspaces that we'd defined really exist. Otherwise, the PowerShell-script will make sure that erroneous build definition isn't triggered by check-ins. A future update of this script might also include emailing the project admins of that particular Team Project about the change of the Build Definition.

Doing a solution like this is pretty simple because the Object Model of TFS is so script friendly.

Real World Reference Stories

In this chapter, real world scenarios are described to demonstrate what happens “out there.” None of the following scenarios are fictitious. Only the company names were made anonymous. The intention of this chapter is not to point a finger in the direction of the one who’s responsible for all the mess, but to show challenges, provide solutions and ask questions which will sooner or later appear in every TFS administrator’s life.

Scenario 1 – Size doesn't matter

Contoso, Ltd used to have a dual server TFS installation for a total of about 20 users. The installation was done rather ad-hoc by an external company without major effort put into planning. The SQL Server used for TFS was also taken to host some other databases that had nothing to do with the TFS installation. Even though this works from a technical point of view, this combination resulted in troubles regarding maintenance due to the single common database server.

After a very short duration of usage, **Contoso, Ltd** decided to move to a single server installation for TFS and to use the underlying SQL Server exclusively for TFS.

The following ways were considered as appropriate to do the migration:

1. Backup the databases and restore them on the new server – due to differences in SQL Server versions, this didn't work.
2. Migration of Source Code and Work Items using [TFS Integration Tools](#)⁹⁹ – this didn't work due to security policy in the local network.
3. Manual migration of sources (Get latest version from "old" server, manual Check-In to "new" server) – if you do this, you would lose version history.
4. Manual Migration of Work Items using Excel.

The customer decided for #3 and #4 and accepted potential loss of data (history/work item links/ attachments), but was happy afterwards, having a stable, fast and easy to maintain installation.

Challenges

The technical challenge in this scenario is the migration from the "old" server to the "new" server without loss of data. The best way depends a lot on the specific scenario. For migration scenarios we recommend that you read the [Upgrade Guidance](#)¹⁰⁰ and [Installation Guidance](#)¹⁰¹. However, for this Team Project Guidance, the challenge would have been to find out what best suits infrastructure for this company *in the first place*. A reinstallation of TFS after a short period should never have been necessary. Good planning should be part of every TFS installation – regardless of whether the installation is big or small.

Lessons learned

- Plan your installation – no matter if you are planning for a small company or an enterprise.
- Think about possible side effects when you share your TFS servers for multiple applications. This includes, for example, version updates and service packs.
- If you are not sure what to do, try to find answers using official installation instructions or guidance documents provided by the ALM Rangers, such as:
 - TFS Installation Guidance
 - Team Project Guidance written by the ALM Rangers (this document)
- Ask TFS consulting experts if you are not sure what to do. There are plenty of companies out there who are willing to help you get your system going. If you're unsure about who to ask, contact Microsoft and ask for a recommendation.
- Keep in mind that money spent for planning will pay out twice later.

⁹⁹ <http://tfsintegration.codeplex.com/>

¹⁰⁰ <http://go.microsoft.com/fwlink/?LinkID=230948>

¹⁰¹ <http://go.microsoft.com/fwlink/?LinkID=231279>

Scenario 2 – Getting complex

Adventure Works is an insurance company having between 200 and 300 developers. They are split into several teams each having between 5 and 25 members. These teams are working on about 50 Team Projects split over two Team Project Collections. Ten of the Team Projects host development of “new stuff” and the rest are focused on maintenance. Newly created Team Projects are mapped to either business projects or to teams.

In source control for each new development, a new branch is created which leads to a branching structure that is 10 or more levels deep.

The TFS infrastructure consists of a separate Application tier and a clustered SQL Server. The server was upgraded several times since the first installation of TFS 2005. In addition to that, six build servers are running. The build servers are shared among Team Projects. No tags are used.¹⁰² This leads to quite long queues on every build controller. To get easier access to data regarding the status of the build environment, custom tools have been created.

Challenges

In this scenario, there are several points to be addressed—not all of them directly connected, but most of them somehow related to the structure of TFS. It’s never easy to point out something as “wrong” from a distance. Still, here are some things to consider:

Lessons Learned

- Is the mapping of Team Projects to Teams optimal or could you decrease complexity using less Team Projects?
- Do you have a branching strategy that you can rely on? If you are not sure or feel it’s too complex, try to follow the ALM Rangers [Branching Guide](#) ¹⁰³.
- Do you need more build servers? Could you improve feedback time and reduce complexity by using additional build agents or by a redesign of your build infrastructure? You might find an answer in the ALM Rangers [Build Customization Guidance](#) ¹⁰⁴.

¹⁰² Note that the use of tags isn’t going to magically make queue times shorter.

¹⁰³ <http://go.microsoft.com/fwlink/?LinkID=230936>

¹⁰⁴ <http://go.microsoft.com/fwlink/?LinkID=230938>

Scenario 3 – New releases every day

Litware, Inc. an ISV with 100 developers, uses a single Team Project Collection to host the development of their products. For every product, a new Team Project is created. Typically, a product has two major releases a year, but every day some patches are released. Each patch usually has its own branch and build. The result is a total of more than 30 active branches, more than 30 active builds and slow build report performance.

The source control structure is flat for about 250 projects, which brings some challenges on the build administration site. A build typically takes 15 minutes of which 5 minutes are spent getting source code (without cleaning), 5 minutes for building and 5 minutes to drop.

The build infrastructure consists of 10 agents running on the same server and one controller. The agents are tagged and tied to branches this way.

Challenges

So what's wrong here? **Nothing!**

As long as it works and you can handle the complexity. But thinking about what might come up in the future, there's always room for improvements:

Lessons Learned

- Think about dispensing the agents on several servers, if you are running into troubles regarding shared OS resources.
- Think about the source structure, following some recommendations in the ALM Rangers [Branching and Merging Guidance](#) ¹⁰⁵.
- Think about if you really need to drop every build—for example, it might be a good idea to throw away build results from a successful gated check-in build to save the time for the file copy and the disk space. It might be a better idea to just keep results of one daily build (for example, Nightbuild) and manually triggered builds.

¹⁰⁵ <http://go.microsoft.com/fwlink/?LinkId=230936>

Scenario 4 – Across the globe

Lucerne Publishing is running a TFS infrastructure consisting of a single Application tier and a single instance database. Lucerne Publishing is located in Europe and hosts around about 300 Team Projects. Most of the Team Projects are kept in a single collection but some of the Team Projects are isolated in their own collection. Lucerne Publishing never had any problems until a project team based in Asia started using build automation. The builds were set up to clean up the local workspace and to execute a “full Get” every time a build was triggered. This led to an enormous decrease of networking performance when single developers in Asia tried to perform a “Get Latest” on their sources.

Lucerne Publishing solved the problem by putting up a central build controller/agent infrastructure which was hosted physically close to the TFS Application tier. This way, it was no longer necessary for the build agent to get sources all across the globe whenever a build was triggered. Whenever the build was done, the build output was sent to a drop location physically close to the team based in Asia.

To improve the developer user experience in Asia, Lucerne Publishing put up a TFS Proxy right next to the developers in Asia. The TFS proxy caches source control data and every “Get” could be executed directly against the proxy.

Challenges

Running an infrastructure all across the globe isn’t easy. TFS brings the option to install a proxy that will be a very good solution to a lot of issues regarding performance in slower networks. But there’s more to consider.

Lessons Learned

- Think about which data is typically being transferred when a developer does his or her daily business. Typically this isn’t necessarily a full “Get” but might be only an “Update Get”.
- Teach your developers to use the “Cloaking” feature Visual Studio / TFS offers. This will reduce execution time of “Get” for the developer by explicit exclusion of files to be transferred.
- Consider using Build Servers that query a TFS proxy instead of the TFS app tier for data. The TFS proxy only caches source code files—all a build server needs is source code files. Seems like a perfect match, doesn’t it?
- Think about physical locations of your infrastructure. It’s always a good idea to stay as close to your team as possible. Running multiple teams across the globe might result in several build servers spread across the globe.
- Think about which data transfer happens most often and where the amount of data being transferred is biggest. Start looking for improvements there. In the scenario described, the builds triggered from Asia happened often and transferred all source code, which makes it a good area for optimization.

Scenario 5 – No time for refactoring

Northwind Traders is a small software company (about 25 developers) that does contract development for huge ISVs. Some of the projects are hosted in-house and **Northwind Traders** has their own TFS infrastructure for these projects. The TFS infrastructure was set up in 2008 and was mainly administrated by the developer team. In the beginning, no clear strategy was being followed regarding Team Project, Team Project Collections and permissions. This resulted in a high number of Team Projects hosted in a single Team Project Collection. Some of the Team Projects were not used productively and have been installed, for example, to find out what another TFS process template brings or to test some modifications. On top of that, almost every new customer order resulted in a new Team Project. Every internal tool development resulted in a new Team Project, too. Even though the company is working within an active directory (AD) and uses AD groups, users were added to Team Projects, user-by-user, not using AD groups. At the time a new developer was hired, he got SharePoint, Reporting and Team Foundation permissions manually, which made the administration very uncomfortable. With the installation of TFS 2010, **Northwind Traders** decided to get rid of all the problems. The company used the upgrade process to create a separate instance of TFS for testing purposes only. The “productive” TFS infrastructure was installed from scratch and relevant old data was imported. In the end, the system had hardly any downtime at all, because the new system was set up before the old system was stopped. AD groups were created, which made handling of new colleagues very convenient. Last but not least, the Team Project structure was redesigned to have a single Team Project for every customer instead of Team Project for every order of the same customer. This significantly reduced the number of Team Projects and made sharing of code or components much easier. For internal projects, a single separate Team Project was created, which hosts all developments of all internal tools.

Challenges

The scenario described here shows how a company gets rid of an unpleasant infrastructure by refactoring concurrently with already required installation efforts. In scenarios like these, where you could continue to work the way you are used to, but feel that something is wrong, there’s always the question of when you will find the time to refactor. Using the upgrade process can be a very effective decision.

Lessons Learned

- The best way to handle situations like these is to avoid them in the first place.
- If you are in such a situation, try to find a good point in time. The upgrade from TFS 2010 to TFS 2012 might suit perfectly. You can run a full test upgrade if you have the chance to work on separate hardware. If you don’t have additional hardware you could try running on virtual machines. On your test installation, you can easily test your refactoring.
- Try to address all pain points that you are experiencing. Some of them – like the usage of AD groups – might seem very small but have a very high impact on administration.
- Keep an eye on the number of Team Projects.
- Keep an eye on the permissions. It’s not a good idea to give administrative permissions in TFS to everyone in your company, even though you might know and like all of the people.
- Create a separate instance of TFS on separate hardware or at least a separate Team Project Collection where people are allowed to “try something out.” You can be much less restrictive with permissions on that system.
- If you are refactoring and restructuring – why not drop data you won’t need any longer? Try to be sure that old data won’t be needed any longer. Even though there are easy ways to upgrade an existing system to the next version without the loss of data, in situations where the old system is messed up, it might be better to start over on a clean system and to import only relevant data, such as the latest version of source code without the history. If you feel that this will probably work for you, it’s a very easy mechanism to clean up your system completely and to get the chance to start over.

Scenario 6 – Disaster by hotfix - Vladimir's DR Experience

Context

- A Russian time zone change created the need for a patch.
- 400+ users worldwide were impacted.
- The Team Project Collection was a 500GB database.
- Twice daily incremental backups and weekly full backups were in place.
- Microsoft supplied a patch to address the issue.

Walkthrough

1. Hotfix for time zone issue related to localization was received from Microsoft.
2. Verified that full backup of AT and DT from previous night were available.
3. Patch applied to production environment.
4. AT partially functioning. Build subsystem stopped working.
5. Issue became a fire ... disaster!

Reaction

1. Discussions between support and Vladimir recommended an uninstall and re-install of the hotfix.
 - Performed however; could not bring TFS back online; AT down.
2. Vladimir decided to re-install TFS AT's OS from scratch.
3. Discussions between support and Vladimir recommended a restore of the configuration database.
4. Application tier was back, but only partially functional.
5. Day 1 passed and operational environment was still not functional.
6. Decided to start using server without build system and fix it later.
7. Check-ins were uncontrolled; recommendation to manage user base.
8. Identified that stored procedure wasn't updated as desired during initial hotfix installation. That change was stored in project collection DB and it wasn't rolled back. It is too late to roll it back because users already did check-ins.
9. Received corrected stored procedure from support and finally fixed the issue.

Lessons and Recommendations

1. Roll back everything.
2. Use a test environment to validate *any* change.
3. Create a checklist of tests to confirm that the patch resolves the issue.
4. Don't allow automatic updates to be applied. Keep control and verify that backups are in place before you apply patches.
5. Communication is critical, especially with geographically remote teams.
6. Have a strategy of how to react before you need it.
7. Test localized environments.
8. Test with computers in different time zones is critical with remote teams.

Outcomes

1. Changed backup plan to be more granular (from twice daily incremental to hourly).
2. Perform periodic testing in staging environment; duplicate the environment frequently.

Scenario 7 – Aborted Warehouse Processing - Thorsten's DR Experience

Error "Smoke"

TF53010: The following error has occurred in a Team Foundation component or extension:

Date (UTC): 19.09.2011 09:14:40

Machine: SV-GMG-DEV01

Application Domain: TfsJobAgent.exe

Assembly: Microsoft.TeamFoundation.Framework.Server, Version=10.0.0.0, Culture=neutral,

PublicKeyToken=b03f5f7f11d50a3a; v2.0.50727 Service Host:

Process Details:

Process Name: TFSJobAgent

Process Id: 2432

Thread Id: 3028

Account name: Domain\TFSSERVICE

Detailed Message: TF221122: An error occurred running job Build Warehouse Sync for team project collection or Team Foundation server DefaultCollection.

Exception Message: A severe error occurred on the current command. The results, if any, should be discarded. (type SqlException) SQL Exception Class: 11 SQL Exception Number: 0 SQL Exception Procedure:

SQL Exception Line Number: 0

SQL Exception Server: DT

SQL Exception State: 0

SQL Error(s):

Exception Data Dictionary:

HelpLink.ProdName = Microsoft SQL Server HelpLink.ProdVer = 10.50.1600 HelpLink.EvtSrc = MSSQLServer

HelpLink.EvtID = 0 HelpLink.BaseHelpUrl = http://go.microsoft.com/fwlink HelpLink.LinkId = 20476

Exception Stack Trace: at System.Data.SqlClient.SqlConnection.OnError(SqlException exception,

Boolean breakConnection)

at System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning()

at System.Data.SqlClient.TdsParser.Run(RunBehavior runBehavior, SqlCommand cmdHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, TdsParserStateObject stateObj)

at System.Data.SqlClient.SqlDataReader.HasMoreRows()

at System.Data.SqlClient.SqlDataReader.ReadInternal(Boolean setTimeout)

at Microsoft.TeamFoundation.Framework.Server.ObjectBinder`1.TryMoveNext()

at Microsoft.TeamFoundation.Build.Server.BuildInformationMerger.TryMergeNext()

at Microsoft.TeamFoundation.Build.Server.CommandQueryChangedBuilds.ContinueExecution()

at Microsoft.TeamFoundation.Build.Server.CommandQueryChangedBuilds.Execute(IList`1 teamProjects, DateTime minChangedTime, BuildStatus statusFilter, IList`1 informationTypes)

at

Microsoft.TeamFoundation.Build.Server.TeamFoundationBuildService.QueryChangedBuilds(TeamFoundationRequestContext requestContext, IList`1 teamProjects, DateTime minChangedTime, BuildStatus statusFilter, IList`1 informationTypes)

at Microsoft.TeamFoundation.Build.Adapter.TeamBuildWarehouseAdapter.ProcessAndUploadBuildData()

at Microsoft.TeamFoundation.Build.Adapter.TeamBuildWarehouseAdapter.MakeDataChanges()

at

Microsoft.TeamFoundation.Warehouse.WarehouseSyncJobExtension`1.MakeDataChanges(TeamFoundationRequestContext requestContext, TeamFoundationJobDefinition jobDefinition, String& resultMessage)

at

Microsoft.TeamFoundation.Warehouse.WarehouseSyncJobExtension`1.RunInternal(TeamFoundationRequestContext requestContext, TeamFoundationJobDefinition jobDefinition, DateTime queueTime, String& resultMessage)

at Microsoft.TeamFoundation.Warehouse.WarehouseJobExtension.Run(TeamFoundationRequestContext requestContext, TeamFoundationJobDefinition jobDefinition, DateTime queueTime, String& resultMessage)

Analysis – "first glance"

At first glance, this error does not look dramatic. It's just a failure in the warehouse processing. The issue does not look very good because we have a failing stored procedure. However, during this phase nothing pointed to a really serious issue. We started investigating the issue. Initially we weren't able to determine the root cause, so we decided that it was not worth further error analysis. Therefore, we simply kicked off a complete rebuild of the warehouse (warehouse database and analysis cube). After a while, the warehouse rebuild aborted, showing an error again. This time things were really getting bad. It wasn't the previous SQL exception. This time, the job

processing aborted, stating that there is a database inconsistency in the collection database. Now that we had the real error message, we were able to address the original issue.

Smoke... then fire

We kicked off a dbcc checkdb command. The following result was returned:

CHECKDB found 0 allocation errors and 488 consistency errors in database 'Tfs_DefaultCollection'.

To make things even worse, the log also showed the following message:

repair_allow_data_loss is the minimum repair level for the errors found by DBCC CHECKDB (Tfs_DefaultCollection, repair_allow_data_loss).

Therefore, we were not able to recover the consistency errors introduced into the collection database. Because we cannot guarantee that a repaired collection database is consistent from a logical point of view, we strongly recommended to roll back the whole collection database in order to get a consistent state again. The client also got in touch with Microsoft support to find out whether a repair is a supported scenario. Microsoft support also disagreed and recommended that we recover the last valid database backup.

Fire... then a raging restore firestorm

Recovering from a backup is a straightforward procedure, even if it requires some time. The real horror trip began when we began to investigate which backup could be considered for the recovery. We figured out that the database inconsistency had been introduced about three or four months ago. The good news was that there was still a tape available on a bank account that was old enough so that we have access to a consistent backup. The bad news was that all the data produced between then and the backup was more or less lost.

Recovery

Next, we recovered the old backup from the tape and restored that last good state of the TFS farm. We also tried to recover at least the last information:

- Head Revision of the source control
- Recent Work Item changes

The client performed backups of the TFS system using a standard SQL backup plan. The really bad part is that the client also configured a regular job to check the consistency of the database as part of the backup maintenance plan. However, nobody monitored the result of the step because all the backups were written properly. Otherwise, they would have realized the error much sooner.

Learning

This case shows once again how critical and essential monitoring server systems really is. This not only includes operational parameters like 'are the TFS web services available' or 'is the SharePoint site online.' it also has to cover the database and the integrity of it.

Appendix

Enabling and using TFS Logging Examples

When you plan for an upgrade, a migration, or even a pristine new environment, it is often invaluable to gather and analyze current and planned traffic with TFS and associated systems. This appendix introduces two of the options for logging clients that are available to you: Visual Studio and Debug View.

Using Visual Studio as Logging Client

One great option is to enable logging as part of the Visual Studio client. Some typical scenarios that you can log and analyze include:

- Analyze traffic and performance of traffic associated with a TFS instance, Team Project Collection and/or Team Project connection.
- Analyze traffic and performance of traffic associated with a check-in or check-out of version control items.

Configuring Visual Studio

1. Close all instances of Visual Studio.
2. Backup the **devenv.exe.config** configuration file that you will be modifying.
3. Edit the **devenv.exe.config** as outlined below for your specific version of Visual Studio.

Visual Studio 2012 Users

Run notepad as an administrator to open the file **C:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\IDE\devenv.exe.config** to enable tracing for Visual Studio 2012.

Configure the following just before the `</configuration>`:

```
<system.diagnostics>
  <switches>
    <add name="TeamFoundationSoapProxy" value="4" />
    <add name="VersionControl" value="4" />
  </switches>
  <trace autoflush="true" indentsize="3">
    <listeners>
      <add name="perfListener"
        type="Microsoft.TeamFoundation.Client.PerfTraceListener,
          Microsoft.TeamFoundation.Client, Version=11.0.0.0,
          Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"/>
      <add name="myListener"
        type="Microsoft.TeamFoundation.TeamFoundationTextWriterTraceListener,
          Microsoft.TeamFoundation.Common, Version=11.0.0.0,
          Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
        initializeData="c:\Temp\VS2011Log.log" />
    </listeners>
  </trace>
</system.diagnostics>
```

Visual Studio 2010 Users

Run notepad as an administrator to open the file **C:\Program Files (x86)\Microsoft Visual Studio 10.0\Common7\IDE\devenv.exe.config** to enable tracing for Visual Studio 2010.

Configure the following just before the </configuration>:

```
<system.diagnostics>
  <switches>
    <add name="TeamFoundationSoapProxy" value="4" />
    <add name="VersionControl" value="4" />
  </switches>
  <trace autoflush="true" indentsize="3">
    <listeners>
      <add name="perfListener"
        type="Microsoft.TeamFoundation.Client.PerfTraceListener,
          Microsoft.TeamFoundation.Client, Version=10.0.0.0, Culture=neutral,
          PublicKeyToken=b03f5f7f11d50a3a"/>
      <add name="myListener"
        type="Microsoft.TeamFoundation.TeamFoundationTextWriterTraceListener,
          Microsoft.TeamFoundation.Common, Version=10.0.0.0, Culture=neutral,
          PublicKeyToken=b03f5f7f11d50a3a"
        initializeData="c:\Temp\VS2010Log.log" />
    </listeners>
  </trace>
</system.diagnostics>
```

Visual Studio 2008 Users

Run notepad as an administrator to open the file **C:\Program Files (x86)\Microsoft Visual Studio 9.0\Common7\IDE\devenv.exe.config** to enable tracing for Visual Studio 2008.

Configure the following just before the </configuration>:

```
<system.diagnostics>
  <switches>
    <add name="TeamFoundationSoapProxy" value="4" />
    <add name="VersionControl" value="4" />
  </switches>
  <trace autoflush="true" indentsize="3">
    <listeners>
      <add name="perfListener"
        type="Microsoft.TeamFoundation.Client.PerfTraceListener,
          Microsoft.TeamFoundation.Client, Version=9.0.0.0, Culture=neutral,
          PublicKeyToken=b03f5f7f11d50a3a"/>
      <add name="myListener"
        type="Microsoft.TeamFoundation.TeamFoundationTextWriterTraceListener,
          Microsoft.TeamFoundation.Common, Version=9.0.0.0, Culture=neutral,
          PublicKeyToken=b03f5f7f11d50a3a"
        initializeData="c:\Temp\VS2008Log.log" />
    </listeners>
  </trace>
</system.diagnostics>
```

4. Launch Visual Studio.

5. Take note of the DialogPerfListener Window, shown in Figure 42, which launches in the background, together with Visual Studio.

The DialogPerfListener window displays two tables of performance data. The top table shows summary statistics for various web services, and the bottom table shows detailed run information for each service call.

WebService	TotalTime	Last	Count	Average	MinTime	MaxTime
GetTeamConfigurationForUser[Services]	242	242	1	242	242	242
GetRegistrationEntries[Services]	194	194	1	194	194	194
GetMetadata2[WorkItemTracking]	159	159	1	159	159	159
Connect[Services]	112	112	1	112	112	112
QueryItemsExtended[VersionControl]	98	94	2	49	44	54
QueryPendingChangeSignature[VersionControl]	54	54	1	54	54	54
QueryNodes[Administration]	36	36	1	36	36	36
QueryResources[Administration]	26	26	1	26	26	26
Connect[Administration]	5	5	1	5	5	5

WebServiceCalled	RunTime	DateTimeStamp	ThreadId
Connect[Services]	112	1/24/2012 6:45:28 PM	6
Connect[Administration]	5	1/24/2012 6:45:28 PM	6
QueryResources[Administration]	26	1/24/2012 6:45:28 PM	6
GetTeamConfigurationForUser[Services]	242	1/24/2012 6:45:28 PM	6
QueryPendingChangeSignature[VersionControl]	54	1/24/2012 6:45:28 PM	7
QueryItemsExtended[VersionControl]	44	1/24/2012 6:45:28 PM	7
QueryNodes[Administration]	36	1/24/2012 6:45:28 PM	6
QueryItemsExtended[VersionControl]	94	1/24/2012 6:45:28 PM	1
GetRegistrationEntries[Services]	194	1/24/2012 6:45:28 PM	5
GetMetadata2[WorkItemTracking]	159	1/24/2012 6:45:28 PM	5

Figure 42 – DialogPerfListener window

NOTE

If you want to see how performance matches up to what others are seeing, you can highlight the data lines and use **Control + C** to copy them to the clipboard. Then you can compare them to the run times for the same project.

6. (Optional) Click Save All to log the data into a text file, as shown in Figure 43.

The DialogPerfListener log file contains the following sections:

Summary

WebService	Total Time	Count	Min Time	Max Time	Average	Last
GetTeamConfigurationForUser[Services]	242	1	242	242	242	242
GetRegistrationEntries[Services]	194	1	194	194	194	194
GetMetadata2[WorkItemTracking]	159	1	159	159	159	159
Connect[Services]	112	1	112	112	112	112
QueryItemsExtended[VersionControl]	98	2	44	54	49	54
QueryPendingChangeSignature[VersionControl]	54	1	36	54	36	54
QueryNodes[Administration]	36	1	26	36	26	36
QueryResources[Administration]	26	1	26	26	26	26
Connect[Administration]	5	1	5	5	5	5

Run Information

```

1 --- Row Index
Connect[Services] --- Web service called
112 --- Run time
1/24/2012 6:45:28 PM --- Start date time stamp
1/24/2012 6:45:28 PM --- End date time stamp
6 --- Thread Id
Normal --- Thread Priority

```

[http://vsalm-test:8080/tfs/rangers%20tfs11%20tpc/Services/v3.0/LocationService.aspx]

at Microsoft.TeamFoundation.Client.Channels.TfsHttpWebRequest.SendRequest()
at Microsoft.TeamFoundation.Client.Channels.TfsHttpRequestChannel.Request(TfsMessage message, TimeSpan timeout)
at Microsoft.TeamFoundation.Client.Channels.TfsHttpClientBase.Invoke(TfsClientOperation operation, Object[] parameters, TimeSpan timeout, Object[]& outputs)
at Microsoft.TeamFoundation.Framework.Client.LocationWebService.Connect(Int32 connectionOptions, Int32 lastChangeId, Int32 features)
at Microsoft.TeamFoundation.Framework.Client.FrameworkServerDataProvider.Connect(ConnectionOptions connectionOptions)
at Microsoft.VisualStudio.TeamFoundation.TeamFoundationHost.AuthenticateWorker.Dowork(Object argument, CancelEventArgs e)
at Microsoft.TeamFoundation.Common.TfsBackgroundWorkerManager.Worker.Dowork(Object sender, DoworkEventArgs e)
at System.ComponentModel.BackgroundWorker.WorkerThreadStart(Object argument)
at System.Runtime.Remoting.Messaging.StackBuilderSink.PrivateProcessMessage(IntPtr md, Object[] args, Object server, Object[]& outArgs)
at System.Runtime.Remoting.Proxies.AsyncWorkerItem.ThreadPoolBack(Object o)
at System.Threading.QueueUserWorkItemCallback.WaitCallback_Context(Object state)
at System.Threading.ExecutionContext.RunInternal(ExecutionContext executionContext, ContextCallback callback, Object state, Boolean preserveSyncCtx)
at System.Threading.ExecutionContext.Run(ExecutionContext executionContext, ContextCallback callback, Object state, Boolean preserveSyncCtx)
at System.Threading.QueueUserWorkItemCallback.System.Threading.ThreadPoolWorkItem.ExecuteWorkItem()
at System.Threading.ThreadPoolWorkQueue.Dispatch()
at System.Threading.ThreadPoolWaitCallback.PerformWaitCallback()

2 --- Row Index
Connect[Administration] --- Web service called
5 --- Run time
1/24/2012 6:45:28 PM --- Start date time stamp
1/24/2012 6:45:28 PM --- End date time stamp
6 --- Thread Id
Normal --- Thread Priority

Figure 43 – DialogPerfListener logfile

Using Debug View as Logging Client

Another option is to monitor TFS connection and processing traffic and use the DebugView application as the logging client. The application can be downloaded from <http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx> or run from <http://live.sysinternals.com/Dbgview.exe>.

The commands are logged as shown in Figure 44 and optionally logged to a text logfile as shown in Figure 45.

Figure 44 – DebugView tool

Figure 45 – DebugView logfile

References

- Microsoft TechNet Article on DebugView: [Debug View](http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx)¹⁰⁶
- Bart Wullem's article on Tracing TFS Data: [Tracing TFS Data](http://bartwullems.blogspot.com/2010/10/tracing-tfs-data.html)¹⁰⁷
- Buck Hodges article on capturing performance information in Visual Studio: [How to Measure Performance Using the Web Service Performance Dialog](http://blogs.msdn.com/b/buckh/archive/2006/09/25/performance-dialog.aspx?wa=wsignin1.0)¹⁰⁸
- Ed Hintz's article on TFS Client Tracking: [TFS Client Tracing](http://blogs.msdn.com/b/edhintz/archive/2007/03/30/tfs-client-tracing.aspx)¹⁰⁹

¹⁰⁶ <http://technet.microsoft.com/en-us/sysinternals/bb896647.aspx>

¹⁰⁷ <http://bartwullems.blogspot.com/2010/10/tracing-tfs-data.html>

¹⁰⁸ <http://blogs.msdn.com/b/buckh/archive/2006/09/25/performance-dialog.aspx?wa=wsignin1.0>

¹⁰⁹ <http://blogs.msdn.com/b/edhintz/archive/2007/03/30/tfs-client-tracing.aspx>

Working with TFS Performance Counters

TFS 2012 has introduced new Performance counters to assist in monitoring performance, which include TFS Services SQL Connections. TFS Services SQL Connections are new in TFS 2012.

As outlined in <http://go.microsoft.com/fwlink/?LinkID=247336> you can use the following counters to analyze and monitor your TFS environment:

TFS Proxy Services

TFS Proxy Services	Description
Current File Downloads/Sec	Current File Downloads/Sec is the rate that files are being downloaded from the proxy service
Current File Downloads	Current File Downloads indicates the number of files currently being downloaded from the proxy service
Total Files in Cache	The total number of files available in the cache
Total Cache Hits	Total number of download requests served from the file cache
Total Download Requests	The total number of download requests that comes to the file cache
Current Cache Size(Bytes)	Current Cache Size in Bytes

Table 35 – Perf Counters: TFS proxy services

TFS Services

TFS Services	Counters for Services in TFS
Total Number of Failed retry sequences	Number of failed retry sequences
Total Number of Throttling Events	Number of Throttling events with SQL
Total Number of SQL Batches	Number of interactions with SQL (number of batches)
Current SQL Executions/Sec	Current SQL Executions is the rate at which SQL Queries are being performed
Current SQL Notification Queries/Sec	Current SQL Notification Queries is the rate at which Queries for SQL notifications are being performed
Current Task Executed/Sec	Current Task Executed is the rate at which tasks are being executed
Active Team Project Collection Service Hosts	Number of active Team Project Collection service hosts
Active Application Service Hosts	Number of active application service hosts
Active Deployment Service Hosts	Number of active deployment service hosts
Current SQL Execution Retries/Sec	Current SQL execution retries is the rate at which SQL command execution is being retried
Current SQL Connection Retries/Sec	Current SQL connection retries is the rate at which SQL connection retries are being attempted
Current SQL Connection Failures/Sec	Current SQL connection failures is the rate at which SQL connection attempts are failing
Average SQL Connect Time	Average time needed to open a new SQL connection
Active SQL Connections	Number of SQL connections in any state used by TFS services
Average Response Time	Average Response Time is the time, on average, that the server took to process a single request

Planning Guide – Appendix

TFS Services	Counters for Services in TFS
Current Requests/Sec	Current Requests/Sec is the rate at which requests are being processed by the server
Current Server Requests	Current Server Requests indicates the number of currently active requests being processed by the server

Table 36 - Perf Counters: TFS services

TFS Version Control

TFS Version Control	Counters for Version Control in TFS
Average Response Time	Average Response Time is the time, on average, that the Version Control service took to process a single request
Current Requests/Sec	Current Requests/Sec is the rate at which requests are being processed by the Version Control Service
Current Server Requests	Current Server Requests indicates the number of currently active requests being processed by the Version Control service
Current File Downloads/Sec	Current File Downloads/Sec is the rate that files are being downloaded from the Version Control service
Current File Downloads	Current File Downloads indicates the number of files currently being downloaded from the Version Control service
Current File Uploads/Sec	Current File Uploads/Sec" description="Current File Uploads/Sec is the rate that files are being uploaded to the Version Control service
Current File Uploads	Current File Uploads indicates the number of files currently being uploaded to the Version Control service

Table 37 - Perf Counters: TFS version control

TFS WorkItem Tracking

TFS WorkItem Tracking	Counters for Work Item Tracking in TFS
Latency Window Starts/Sec	Latency Window Starts/Sec is the rate at which server responses initiate latency time window per user - used if replication is enabled
Write Access Elevations/Sec	Write Access Elevations/Sec is the rate at which requests are elevated to Write access
ReadLatest Access Elevations/Sec	ReadLatest Access Elevations/Sec is the rate at which requests are elevated to ReadLatest access
Active GetQueryAccessControlList Requests	The number of access control list query requests currently executing
Active GetStoredQueries Requests	The number of stored queries requests currently executing
Active GetStoredQuery Requests	The number of stored query requests currently executing
Active GetMetadata Requests	The number of cache updates currently executing
Active Update Requests	The number of updates currently executing
Active GetWorkitem Requests	The number of work Item requests currently executing
Active Paging Requests	The number of paging requests currently executing
Active Query Requests	The number of queries currently executing

Table 38 - Perf Counters: TFS work Item tracking

TFS Lab Management

TFS Lab Management	TFS Lab Management Performance Counters
Current Requests	Number of Lab Management requests currently active inside server
Requests/Sec	Number of Lab Management requests processed per second
Current Lab Environment Creations	Number of lab environments TFS is creating on this server
Current Operations	Number of lab environment or template operations (create, start, stop, snapshot, delete, and so forth) in progress on this server
Total Operations	Number of lab environment or template operations (create, start, stop, checkpoint, delete, and so forth) completed since the last reboot. This number also includes unsuccessful operations
Total Lab Environment Creations	Number of lab environments created since the last reboot
Total Lab Environment Creation Failures Due To Lack Of Resources	Number of unsuccessful lab environment creations because of a lack of resources since the last reboot
Powershell Cmdlets/Sec	Number of PowerShell cmdlets executed per second
Runspaces Created	Number of runspace created
Kvp Data Cache : # entries	Number of entries in Kvp data cache
Kvp Data Cache : hit ratio	Hit Ratio of Kvp cache
Kvp Data Cache : hit ratio base counter	Hit Ratio Base counter of Kvp cache
Kvp Data Cache : # trims	Number of entries removed either invalid or to make space for others in Kvp cache
VMMS Cache : # entries	Number of entries in the SCVMM cache
VMMS Cache : hit ratio	Hit ratio of the SCVMM cache
VMMS Cache : hit ratio base counter	Hit ratio base counter of the SCVMM cache
VMMS Cache : # trims	Number of entries removed either they were not valid or to allocate space for other entries in the SCVMM cache
CS Cache : # entries	Number of entries in 'computer system objects' cache
CS Cache : hit ratio	Hit Ratio of 'computer system objects' cache
CS Cache : hit ratio base counter	Hit Ratio Base counter of 'computer system objects' cache
CS Cache : # trims	Number of entries removed either invalid or to make space for others in 'computer system objects' cache
VMM Object Cache : VM & Template : # entries	Number of entries in the VM and Template cache
VMM Object Cache : Location : # entries	Number of entries in the Host, Host Group, and Library Share caches
VMM Object Cache : Snapshot : # entries	Number of entries in Snapshot cache
VMM Object Cache : Task : # entries	Number of entries in Task cache
VMM Object Cache : Profile : # entries	Number of entries in the Hardware profile, OS profile, and User role profile caches

Table 39 - Perf Counters: TFS Lab Management

DR Recovery Walkthroughs

Walkthrough: Planning the ideal DR environment for TFS

Planning

Step	Description	Details	Done
1	Analyze Server Strategies	Refer to page 9 to decide which server strategy, for example on-premises, is most ideal for the requirements and IT environment.	<input type="checkbox"/>
2	Analyze DR Strategies	Using Table 19 on page 46, determine the best DR strategies for the TFS Server strategy.	<input type="checkbox"/>
3	Select a DR Strategy	See the following tables, based on the selected DR Strategy: <ul style="list-style-type: none"> Table 41 – Walkthrough: Planning the ideal DR environment for TFS – Simple Strategy, page 80. Table 42 – Walkthrough: Planning the ideal DR environment for TFS – Backup Strategy, page 81. Table 43 – Walkthrough: Planning the ideal DR environment for TFS – Standby Strategy, page 81. Table 44 – Walkthrough: Planning the ideal DR environment for TFS – Cluster Strategy, page 82. 	<input type="checkbox"/>
4	Document DR Strategy	Document the DR strategies, reasoning and recovery processes.	<input type="checkbox"/>
5	Infrastructure Diagram	Create an infrastructure / topology diagram, including machine names and roles, that can be used as an invaluable reference during disaster recovery.	<input type="checkbox"/>
6	Evaluate and refine the DR Strategy	Evaluate the planned DR environment, refine the recovery process, and train the support staff.	<input type="checkbox"/>
7	Evangelize your DR Strategy	Raise the awareness of the selected DR Strategy and ensure that the escalation process and the contact information are accessible to all stakeholders.	<input type="checkbox"/>

Table 40 – Walkthrough: Planning the ideal DR environment for TFS

Simple Strategy

When you opt for the simple strategy, historical data such as version control or work item history, is **not** important. This strategy is typically used in evaluation, prototyping, or training environments, with limited, if any, service level agreements and disaster recovery requirements.

Option	Description	Details	Done
1	Physical environment	Consider a single server topology to keep installation, configuration, and administration as simple as possible.	<input type="checkbox"/>
2	Virtualized environment	The use of a virtual machine allows recovery from snapshots or backups and offers straightforward disaster recovery.	<input type="checkbox"/>

Table 41 – Walkthrough: Planning the ideal DR environment for TFS – Simple Strategy

Backup Strategy

The objective of the backup strategy is to prevent data loss due to hardware and/or software failure.

Step	Description	Details	Done
1	Understand TFS Server Backup and Restore	Read Backing up and Restoring Your Deployment ¹¹⁰ .	<input type="checkbox"/>
2	Define a backup and restore plan	Define a backup, restore, and validation plan for your deployment topology for TFS. This should be applicable for SharePoint Products, SQL Server Reporting, SQL Server Analysis Services, TFS Build, and VS Test Controllers.	<input type="checkbox"/>
3	Do you need to restore to a different TFS instance?	To be able to restore to a different TFS instance, you need to back up and restore the Tfs_Configuration database as well. If you need to move a collection to another server, the recommended way is to: <ul style="list-style-type: none"> (a) Stop and detach your Team Project Collection, which copies the shared data from Tfs_Configuration database into the collection database and produces a portable collection. (b) Back up the collection. (c) Re-attach Team Project Collection. See https://gist.github.com/2865415 for a handy script that automates detach, back up, and re-attach. This process takes recovery time, which results in outages for users of the affected collection. Account for this recovery time in your DR planning and service level agreements.	<input type="checkbox"/>
4	Do you need a “latest” shadow version control copy?	Often software development teams will ask for a shadow of the latest version control data on a file server to use for quick reference or backup during backup outage times. You can automate the “get latest” using PowerShell and create a shadow update process, which can optionally be run before scheduled outages. See Get latest files from TFS using PowerShell ¹¹¹ for an example.	<input type="checkbox"/>
5	Validate ²	Validate the backup, restore, and recovery processes before switching on production and whenever an infrastructure change is planned.	<input type="checkbox"/>

Table 42 – Walkthrough: Planning the ideal DR environment for TFS – Backup Strategy

Standby Strategy

The objective of the standby strategy is to avoid outages due to the failure of a TFS Application-tier server.

Option	Description	Details	Done
1	Warm standby AT Server	Review Activating a Fail-Over Application-Tier Server ¹¹² , which covers the “how to” set up and activate a warm standby AT server. When the primary server fails, you have to complete manual activation steps. This implies that there will be an activation outage that ranges from brief to substantial.	<input type="checkbox"/>
2	Load balanced AT Server	Review How to: Create a Team Foundation Server Farm (High Availability) ¹¹³ , which covers the use of Network Load Balancing ¹¹⁴ . Because there is no need for manual intervention, there will be no outage when an AT server fails.	<input type="checkbox"/>

Table 43 – Walkthrough: Planning the ideal DR environment for TFS – Standby Strategy

¹¹⁰ [http://msdn.microsoft.com/en-us/library/bb552295\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/bb552295(v=vs.100).aspx)

¹¹¹ <http://blogs.infosupport.com/getting-latest-files-from-tfs-using-powershell/>

¹¹² [http://msdn.microsoft.com/en-us/library/ms252486\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/ms252486(v=VS.90).aspx)

¹¹³ <http://msdn.microsoft.com/en-us/library/ee259689.aspx>

¹¹⁴ <http://technet.microsoft.com/en-us/library/cc770558.aspx>

Cluster Strategy

The objective of the cluster strategy is to avoid outages due to the failure of a TFS Data-tier server. You can consider mirroring or clustering to handle failover.

Option	Description	Rating ¹¹⁵	Details	Done
1	SQL Mirroring	★ ★ ★ ★	<ul style="list-style-type: none"> – Automatic mirror failover not supported. + Lower cost than clustering. <p>Database mirroring maintains a copy of TFS databases on a principal TFS DT synchronized with the databases on the mirroring TFS DT. If the principal TFS DT fails, you can manually swap the roles and promote the backup to the primary TFS DT. Read here ¹¹⁶ for more information.</p>	<input type="checkbox"/>
2	Active-Passive Cluster	★ ★ ★ ★	<ul style="list-style-type: none"> – Cost is higher than mirroring. – TFS AT cannot reside on a cluster server. + Automatic failover provided by clustering. <p>A SQL Server cluster is a set of servers configured to appear as a single server, with an active and a passive (standby) node. Clustering allows active failover for unexpected failures and planned maintenance. Read here ¹¹⁷ for more information.</p>	<input type="checkbox"/>
3	Active-Active Cluster	★ ★ ★ ★	<ul style="list-style-type: none"> – Each node must be capable of handling the load of all SQL Server instances during a SQL Server cluster node failover. + Use each node to run one or more SQL Server instances. <p>Split TFS and SharePoint data onto separate active node servers, with one passive standby server. Read here ¹¹⁸ for more information.</p>	<input type="checkbox"/>
4	Active-Active-Passive Cluster	★ ★ ★ ★	<ul style="list-style-type: none"> – Additional passive server(s) increase cost. + Always a Passive node available for failover scenarios. <p>Split TFS and SharePoint data onto separate active node servers, with one passive standby server. Read here ¹¹⁹ for more information.</p>	<input type="checkbox"/>
5	Multi-Subnet Failover Clustering "SQL Always On"	★ ★ ★ ★	<ul style="list-style-type: none"> – SQL Server 2012 or later versions only. + Ability to failover to a different datacenter. <p>Configure SQL Server with failover cluster nodes in different subnets within the same Active Directory domain.</p> <p>Note that TFS 2010 does not support the full AlwaysON technology in SQL 2012 – it only supports basic SQL usage. Read here ¹²⁰ for more information.</p>	<input type="checkbox"/>

Table 44 – Walkthrough: Planning the ideal DR environment for TFS - Cluster Strategy

¹¹⁵ Unofficial failover rating, to indicate that clustering is preferred over mirroring

¹¹⁶ [http://msdn.microsoft.com/en-us/library/aa980644\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/aa980644(v=vs.90).aspx)

¹¹⁷ [http://msdn.microsoft.com/en-us/library/aa980644\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/aa980644(v=vs.90).aspx)

¹¹⁸ [http://msdn.microsoft.com/en-us/library/aa980644\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/aa980644(v=vs.90).aspx)

¹¹⁹ [http://msdn.microsoft.com/en-us/library/aa980644\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/aa980644(v=vs.90).aspx)

¹²⁰ <http://msdn.microsoft.com/en-us/library/ff877884.aspx>

Farms – Learning

DNS Aliases

- Use as many DNS alias names as possible. This lets you scale in new systems while keeping the original system online. After the new system is prepared, you can transparently switch to the new server without downtime, or with as little downtime as possible.
- If a proxy or a TFS DT fails you can simply point the DNS alias to the new system and off you go again. The same is true for backend systems like the warehouse or analysis DB.

Example Infrastructure

The following diagram shows a possible DR environment. Although these are important part of your disaster recovery plan, note that SQL Reporting and Analysis services, Test and Build servers, and shared network storage failover are **not** shown in the diagram.

Mirroring SQL Server data might be useful in this scenario to mirror the data to a different physical location, for example, another physical datacenter.

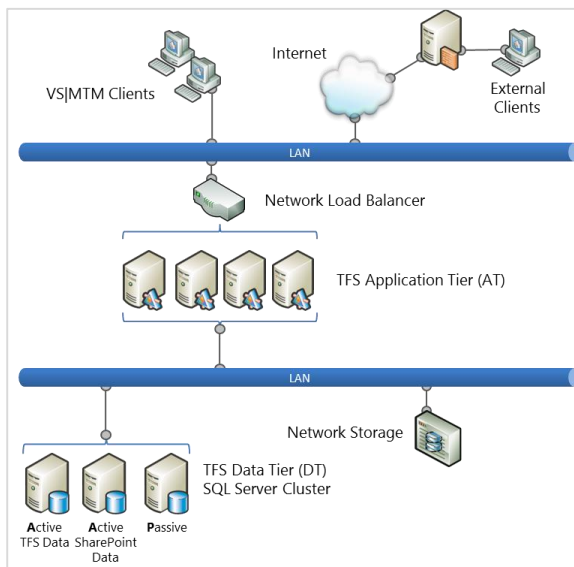


Figure 46 - Example of a possible disaster recovery environment

Walkthrough: DR Recovery - Complete Failure (aka Datacenter fire or meteor strike)

Walkthrough Checklist Single Server installation

Step	Description	Reference Link(s)	Done
1	Check if you have a disaster recovery guide for your organization.		<input type="checkbox"/>
2	Check if you have backup of your database(s) available.		<input type="checkbox"/>
3	If you have a secondary site in place, follow the walkthrough on page 87.		<input type="checkbox"/>
4	If you don't have a secondary site in place, install the new server. The guide includes: <ul style="list-style-type: none"> • Prepare the new hardware. • Restore the databases. • Install and configure TFS. • Reconnect services and users. 	See Restore a Single-Server Deployment to New Hardware (2010) ¹²¹ . See Restore a Single-Server Deployment to New Hardware (2012) ¹²² .	<input type="checkbox"/>
5	Make sure that the DNS entry points to the new TFS server. If no DNS entry exists, you need to inform the TFS users of the new TFS location.	See Configure TFS to use FQDN ¹²³ .	<input type="checkbox"/>

Table 45 – Walkthrough: Single Server installation

Walkthrough Checklist Multi Server installation

Step	Description	Details	Done
1	Check if you have a disaster recovery guide for your organization.		<input type="checkbox"/>
2	Check if you have backup of your database(s) available.		<input type="checkbox"/>
3	If you have a secondary site in place, follow the walkthrough on page 87.		<input type="checkbox"/>
4	If you don't have a secondary site in place, start recovering all servers, by following = the walkthrough on page 85.		<input type="checkbox"/>
5	Recover the TFS Application tier by following the walkthrough on page 87.		<input type="checkbox"/>
6	If you are using SharePoint servers, recover them by following the walkthrough on page 95.	See Backup and recovery for SharePoint Server 2010 ¹²⁴ . See Backup and recovery for SharePoint Server 2013 ¹²⁵ . See Protecting and restoring the farm (Windows SharePoint Services 3.0) ¹²⁶ .	<input type="checkbox"/>
7	If you are using Proxy Servers, recover them by following the walkthrough on page 91.		<input type="checkbox"/>

¹²¹ <http://msdn.microsoft.com/en-us/library/ff459215.aspx>

¹²² <http://msdn.microsoft.com/en-us/library/hh529826.aspx>

¹²³ [http://msdn.microsoft.com/en-us/library/cc752948\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/cc752948(v=VS.90).aspx)

¹²⁴ [http://technet.microsoft.com/en-us/library/ee662536\(v=office.14\).aspx](http://technet.microsoft.com/en-us/library/ee662536(v=office.14).aspx)

¹²⁵ [http://technet.microsoft.com/en-us/library/ee662536\(v=office.15\).aspx](http://technet.microsoft.com/en-us/library/ee662536(v=office.15).aspx)

¹²⁶ [http://technet.microsoft.com/en-us/library/cc287741\(office.12\).aspx](http://technet.microsoft.com/en-us/library/cc287741(office.12).aspx)

Step	Description	Details	Done
8	If you are using Build Servers, recover them by following the walkthrough on page 92.		<input type="checkbox"/>

Table 2 – Walkthrough: Multi-Server installation

Walkthrough: DR Recovery - SQL Server Dies (DT Failure)

SYMPTOMS

The disaster could be a physical hardware failure or it could be data corruption. Symptoms include error messages that indicate data corruption, hardware related error messages on the hosting server, or an inability to connect to any of the TFS databases.

DT Recovery – Single Server Restoration, New Hardware

Use the following checklist as guidance when all server components are deployed on a single physical server.

Step	Description	Reference Link(s)	Done
1	Prepare the new hardware.	See Prepare the New Hardware ¹²⁷ .	<input type="checkbox"/>
2	Restore the databases.	See Restore the Databases ¹²⁸ .	<input type="checkbox"/>
3	Install and configure TFS	See Install and configure Team Foundation Server ¹²⁹ .	<input type="checkbox"/>
4	Reconnect services and users.	See Reconnect Services and Users ¹³⁰ .	<input type="checkbox"/>

Table 46 – Walkthrough: single-server restoration

DT Recovery – Restore data to the same location

Use the following checklist as guidance for restoring TFS data to the same hardware. This same procedure may be used for either a single-server or multiple-server deployment.

Step	Description	Reference Link(s)	Done
1	Verify permissions.	See Required Permissions ¹³¹ .	<input type="checkbox"/>
2	Stop services.	See Stop Services that Team Foundation Server Uses ¹³² .	<input type="checkbox"/>
3	Restore TFS databases.	See Restore Team Foundation Databases ¹³³ .	<input type="checkbox"/>
4	Update service accounts.	See Update All Service Accounts ¹³⁴ .	<input type="checkbox"/>
5	Rebuild the warehouse.	See Restore the Warehouse ¹³⁵ .	<input type="checkbox"/>
6	Restart TFS services.	See Restart Services that Team Foundation Server Uses ¹³⁶ .	<input type="checkbox"/>

¹²⁷ <http://msdn.microsoft.com/en-us/library/hh529825.aspx>

¹²⁸ <http://msdn.microsoft.com/en-us/library/hh529829.aspx>

¹²⁹ <http://msdn.microsoft.com/en-us/library/hh529828.aspx>

¹³⁰ <http://msdn.microsoft.com/en-us/library/hh529824.aspx>

¹³¹ <http://msdn.microsoft.com/en-us/library/ms252458.aspx#RequiredPermissions>

¹³² <http://msdn.microsoft.com/en-us/library/ms252458.aspx#StopServices>

¹³³ <http://msdn.microsoft.com/en-us/library/ms252458.aspx#RestoreDatabases>

¹³⁴ <http://msdn.microsoft.com/en-us/library/ms252458.aspx#UpdateAccounts>

¹³⁵ <http://msdn.microsoft.com/en-us/library/ms252458.aspx#RebuildDataWarehouse>

¹³⁶ <http://msdn.microsoft.com/en-us/library/ms252458.aspx#RestartServicesForTFS>

Planning Guide – Appendix

Step	Description	Reference Link(s)	Done
7	Refresh client cache.	See Refresh the Caches on Client Computers ¹³⁷ .	<input type="checkbox"/>

Table 47 – Walkthrough: Restore Data to the same location, multiple servers

DT Recovery – Restore data to a different location

Use the following checklist as guidance for restoring TFS data on multiple servers to new hardware.

Step	Description	Reference Link(s)	Done
1	Verify permissions.	See Required Permissions ¹³⁸ .	<input type="checkbox"/>
2	Install and configure SQL Server.	See Install and Configure SQL Server on new hardware ¹³⁹ .	<input type="checkbox"/>
3	Stop services.	See Stop Services that Team Foundation Server Uses ¹⁴⁰ .	<input type="checkbox"/>
4	Restore Team Foundation databases.	See Restore Team Foundation Databases ¹⁴¹ .	<input type="checkbox"/>
5	Redirect SharePoint.	See Redirect SharePoint Products to the New Location of the Content Database ¹⁴² .	<input type="checkbox"/>
6	Modify Reporting Service configuration.	See Change the Database in Reporting Services Configuration Manager ¹⁴³ .	<input type="checkbox"/>
7	Prepare SQL Server.	See Prepare the New SQL Server or Instance for Team Foundation Server ¹⁴⁴ .	<input type="checkbox"/>
8	Modify the database owner.	See Change the Ownership of the Restored Databases ¹⁴⁵ .	<input type="checkbox"/>
9	Redirect TFS to restored Project Collections.	See Redirect Team Foundation Server to Remote Collection Databases ¹⁴⁶ .	<input type="checkbox"/>
10	Update service Accounts.	See Update All Service Accounts ¹⁴⁷ .	<input type="checkbox"/>
11	Register restored databases.	See Register the Location of the Restored Databases ¹⁴⁸ .	<input type="checkbox"/>
12	Rebuild the warehouse.	See Restore the Warehouse ¹⁴⁹ .	<input type="checkbox"/>
13	Clear the Application tier server Cache.	See Clear Data Cache on Server ¹⁵⁰ .	<input type="checkbox"/>
14	Restart TFS services.	See Restart Services that Team Foundation Server Uses ¹⁵¹ .	<input type="checkbox"/>
15	Clear client cache.	See Refresh the Caches on Client Computers ¹⁵² .	<input type="checkbox"/>

Table 48 – Walkthrough: Restore data to different location, multiple servers

¹³⁷ <http://msdn.microsoft.com/en-us/library/ms252458.aspx#RefreshDataCache>

¹³⁸ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RequiredPermissions>

¹³⁹ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#InstallAndConfigure>

¹⁴⁰ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#StopServices>

¹⁴¹ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RestoreDB>

¹⁴² <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RedirectSPT>

¹⁴³ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#ChangeSQLRS>

¹⁴⁴ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#ConfigNewSQL>

¹⁴⁵ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#ChangeOwnership>

¹⁴⁶ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RedirectQLRTPC>

¹⁴⁷ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#UpdateNetworkService>

¹⁴⁸ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RegisterDB>

¹⁴⁹ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RestoreWarehouse>

¹⁵⁰ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#ClearData>

¹⁵¹ <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RestartServices>

¹⁵² <http://msdn.microsoft.com/en-us/library/ms252516.aspx#RefreshDataCache>

Walkthrough: DR Recovery - AT Failure

NOTE

Sometimes it's easier to switch to a new Application tier than to fix a corrupt Application tier.

Implementing a new Application-tier server is recommended in case of hardware failures on disks or software issues like viruses. Because the Application tier holds just temporary cache data files, you **don't lose** data, with the exception of log files, when you start over on a new system.

It is recommended to always know where the media to rebuild the Application Tier is. This reduces the time of getting your environment back up and running. Having the pressure of "where's the media" is something you should take out of the equation.

The following walkthrough summarizes the implementation of a new Application tier (AT) server. It's assumed that the server is a dedicated AT server and that it is not hosting other servers such as Reporting or Build services.

Step	Description	Reference Link(s)	Done
1	Verify the system requirements and server considerations.	See System Requirements for Team Foundation Server ¹⁵³ .	<input type="checkbox"/>
2	Re-install Team Foundation Application tier.	See Restore an Application-Tier Server ¹⁵⁴ .	<input type="checkbox"/>
3	Update the Team Foundation Application tier to pre-disaster patch level.	See Latest Updates for Team Foundation Server 2012 ¹⁵⁵ .	<input type="checkbox"/>
4	Configure an isolated and manageable group of users who were impacted by the failure and verify that TFS is available and fully functional.		<input type="checkbox"/>
5	If a Network Load Balancing solution is used, reconfigure it to include the new TFS AT.	How to: Create a Team Foundation Server Farm (High Availability) ¹⁵⁶ .	<input type="checkbox"/>
6	Alternatively, update the DNS entry for TFS to direct users to the new TFS AT location.	Configure TFS to use FQDN ¹⁵⁷ . Configuring NLB for TFS 2010 ¹⁵⁸ .	<input type="checkbox"/>
7	Alternatively, inform all affected users of new TFS location if the previous DNS name was not re-used.		<input type="checkbox"/>

Table 1 - Walkthrough: Set up a new Application tier

Walkthrough: DR Recovery - Failover to a secondary site

Overview

Failing over to a secondary site is a disaster recovery scenario that is viewed as an extreme situation. This failover has to be well thought out because you have to have a plan of returning to the primary site with any updates made in the secondary site. There are many conditions that need to be considered:

¹⁵³ <http://msdn.microsoft.com/en-us/library/vstudio/dd578592.aspx>

¹⁵⁴ <http://msdn.microsoft.com/en-us/library/dd793167.aspx>

¹⁵⁵ <http://social.msdn.microsoft.com/Forums/en-US/tfsadmin/thread/33034618-778d-423c-9cca-1b4b6edd71fd>

¹⁵⁶ <http://msdn.microsoft.com/en-us/library/ee259689.aspx>

¹⁵⁷ [http://msdn.microsoft.com/en-us/library/cc752948\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/cc752948(v=VS.90).aspx)

¹⁵⁸ <http://blogs.msdn.com/b/tfssetup/archive/2010/11/12/configuring-nlb-for-tfs-2010.aspx>

- Is the secondary environment completely referenceable with different URLs?
- How will the data from the primary environment be transferred and incorporated into the secondary environment?
- How will the primary environment be updated with the secondary environment's updates when you switch back to the primary environment?
- Can the secondary environment be validated and tested while the primary environment is available?
- Is patching of the secondary environment co-ordinated with patches that have been applied to the primary environment?
- How will use of the secondary environment be communicated and then how will you revert back to the primary environment?
- How will you manage dependent components like build and proxy servers outside of the primary site's environment?

You should determine the amount of effort from all parties involved for what it will take to fail over and use this in identifying the criteria to make that call in going to the secondary site.

- Review the business requirements for going to the secondary site.
- Determine all necessary participants (database, development, end users, infrastructure and testers) and have confirmation of availability if a failover is required.
- Determine the time required to prepare the secondary site. Depending on the location and timeliness of getting the primary site's data, this could be a time-consuming task itself.
- Test the move to the secondary site and track the time.

Failing over to secondary site is not a trivial task, so documenting, planning, and testing are key to having a high level of confidence. Ownership of tasks should be established you should test your plan at least once a year.

Walkthrough Checklist

With this walkthrough, these are the conditions:

- An extreme condition has been met and the primary site is going to be unavailable for at least a month.
- The primary and secondary sites are in the same domain. This ensures that the domain remains active.
- The primary site was a single application tier and the secondary site is a single application tier.
- Both the primary site and the secondary site have a warehouse and an analysis database data tier.
- The secondary site is enabled from the last failover test.
- The primary site's TFS databases are on disk images. These images must be ready-to-use TFS databases and be available to the secondary site within four hours.
- Proxy and build servers exist outside of the primary site. The secondary site doesn't have build or proxy servers.
- SharePoint is used for project portals.
- Reporting Services is used for reporting.
- The same patches have been applied to both the primary and the secondary sites.
- All service accounts are domain accounts and are the same for both the primary and secondary sites.
- The participants in the failover process know what activities they've been assigned.
- Activities are documented and tasks are managed by a central person. This person tracks all the tasks that must be completed before the secondary site is pronounced ready to use.

Step	Description	Details	Done
1	Determine Point in Time of Database Images	Identify when the data images are from and initiate the process of making them available to the secondary site's database environment.	<input type="checkbox"/>
2	Communicate Failing Over to Secondary Site	Alert teams that the primary site is unavailable and that any changes after the point in time from step 1 will have to be reapplied in the secondary site.	<input type="checkbox"/>

Planning Guide – Appendix

Step	Description	Details	Done
3	Prepare Secondary Site	<p>Stop the secondary sites TFS related services:</p> <ul style="list-style-type: none"> • SharePoint timer service via the Services interface. • Reporting Services service via the Services interface. • Warehouse processing via the TFS Administration console. • Quiesce TFS from an elevated command prompt using the TFSServiceControl quiesce command. <p>Detach TFS_Configuration, Team Project Collection database and SharePoint content databases.</p>	<input type="checkbox"/>
4	Attach Data Images to Secondary Site	<p>Attach TFS_Configuration, Team Project Collection database and SharePoint content databases that have been presented as a different drive letter than what was used before.</p> <p>Note: The TFS Warehouse and Analysis Services databases will be rebuilt and do not need to be attached.</p>	<input type="checkbox"/>
5	Sync New Secondary Site Databases With Secondary Site Environment	<p>Run attach commands.</p> <p>TFS 2012: Change the service id of the primary site's databases via tfssconfig commands:</p> <ul style="list-style-type: none"> • Changeserverid • Registerdb 	<input type="checkbox"/>
6	Validate Connections	<p>To validate that TFS databases are properly mapped, you can query the configuration database to see the connection strings that TFS will use. Run this query: <code>SELECT * FROM Tfs_Configuration.dbo.tbl_ServiceHost</code></p>	<input type="checkbox"/>
7	Start Connections	<p>Start the secondary sites TFS related services:</p> <ul style="list-style-type: none"> • SharePoint timer service via the Services interface. • Reporting Services service via the Services interface. • Unquiesce TFS from an elevated command prompt. • Warehouse processing via the TFS Administration console. <p>Confirm Collections in the TFS Admin Console are showing online.</p>	<input type="checkbox"/>
8	Review Logs	Check AT and DT Windows and SQL logs for messages.	<input type="checkbox"/>
9	Validate Secondary Site Connectivity	Designated development tester will confirm that secondary URI is accessible in Visual Studio.	<input type="checkbox"/>
10	Update Build and Proxy Server to Use Secondary Site	In the Build Server's TFS Admin Console, change the connection to the secondary site's URI. In the Proxy Server's TFS Admin Console, change the connection to the secondary site's URI.	<input type="checkbox"/>
11	Validate Build and Proxy Server Usability	Designated development tester will confirm that reconfigured Build and Proxy servers are usable.	<input type="checkbox"/>
12	Communicate Availability of Secondary Site	Gradually inform teams that the secondary site is available.	<input type="checkbox"/>
13	Monitor Secondary Site	Designated TFS administrator will monitor event logs for connections that refer back to the primary site.	<input type="checkbox"/>

Table 49 – Walkthrough: Failingover to Secondary Site

Getting Back to the Primary Site

When the primary site is once again operational, you begin to reverse the process that took you to the secondary site.

Planning Guide – Appendix

Step	Description	Details	Done
1	Determine Point in Time of Database Images	Identify when you want to take the data images from the secondary site and make them available to the primary site's database environment.	<input type="checkbox"/>
2	Communicate Plan to Return to Primary Site	Co-ordinate an appropriate time with teams for bringing the primary site back on-line.	<input type="checkbox"/>
3	Stop the Secondary Site	<p>Stop the secondary sites TFS related services:</p> <ul style="list-style-type: none"> • SharePoint timer service via the Services interface. • Reporting Services service via the Services interface. • Warehouse processing via the TFS Administration console. • Quiesce TFS from an elevated command prompt using the TFSserviceControl quiesce command. <p>Detach TFS_Configuration, Team Project Collection database and SharePoint content databases.</p>	<input type="checkbox"/>
4	Prepare Primary Site	<p>Stop the primary sites TFS related services:</p> <ul style="list-style-type: none"> • SharePoint timer service via the Services interface. • Reporting Services service via the Services interface. • Warehouse processing via the TFS Administration console. • Quiesce TFS from an elevated command prompt. <p>Detach TFS_Configuration, Team Project Collection database and SharePoint content databases.</p> <p>Bring the disk images back and make them available to the primary site's database server.</p>	<input type="checkbox"/>
5	Attach Data Images to Primary Site	<p>Attach TFS_Configuration, Team Project Collection database and SharePoint content databases that have been presented as a different drive letter than what was used before.</p> <p>Note: The TFS Warehouse and Analysis Services databases will be rebuilt and do not need to be attached.</p>	<input type="checkbox"/>
6	Sync Primary Site Databases With Primary Site Environment	<p>Run attach commands.</p> <p>TFS 2012: Change the service id of the primary site's databases via tfscfg commands:</p> <ul style="list-style-type: none"> • Changeserverid • Registerdb 	<input type="checkbox"/>
7	Validate Connections	<p>To validate that TFS databases are properly mapped, you can query the configuration database to see the connection strings that TFS will use. Run this query:</p> <pre>SELECT * FROM Tfs_Configuration.dbo.tbl_ServiceHost</pre>	<input type="checkbox"/>
8	Start Connections	<p>Start the primary sites TFS related services:</p> <ul style="list-style-type: none"> • SharePoint timer service via the Services interface, • Reporting Services service via the Services interface, • Unquiesce TFS from an elevated command prompt, • Warehouse processing via the TFS Administration console, <p>Confirm Collections in the TFS Admin Console are showing online.</p>	<input type="checkbox"/>
9	Review Logs	Check AT and DT Windows and SQL logs for messages,	<input type="checkbox"/>
10	Validate Primary Site Connectivity	Designated development tester will confirm that secondary URI is accessible in Visual Studio.	<input type="checkbox"/>

Planning Guide – Appendix

Step	Description	Details	Done
11	Update Build and Proxy Server to Use Primary Site	In the Build Server's TFS Admin Console, change the connection to the primary site's URI. In the Proxy Server's TFS Admin Console, change the connection to the primary site's URI.	<input type="checkbox"/>
12	Validate Build and Proxy Server Usability	Designated development tester will confirm that reconfigured Build and Proxy servers are usable.	<input type="checkbox"/>
13	Communicate Availability of Primary Site	Gradually inform teams that the primary site is available.	<input type="checkbox"/>
14	Monitor Primary Site	Designated TFS administrator will monitor event logs for connections that refer back to the secondary site.	<input type="checkbox"/>

Table 2 – Walkthrough: Going Back to Primary Site

Walkthrough: DR Recovery - Proxy Failure

SYMPTOMS

A dedicated group of users has issues on working with TFS Source Control; the impact could range from slow source control responses to an unreachable TFS.

DR Investigation - Investigation Walkthrough

Area	Solution	Involved personas	Checked
Bypass proxy in use	<p>Questions to ask ...</p> <ul style="list-style-type: none"> Do all reported symptoms belong to user(s) using a proxy? Did they all use the same proxy? Do they all come from the same location? <p>Deactivate the use of proxy on one of the reported clients (in Visual Studio go to Tools, Options, Source Control, and then Visual Studio Team Foundation Server). If the problem seems to be solved by this, you can instruct all the users to bypass the proxy until issue is resolved.</p>	TFS Expert, End user	<input type="checkbox"/>
Service Account	Confirm the password still works and the proxy service account is not disabled. If it is disabled, enable it. If password has changed, reconfigure the proxy and retype the password.	TFS Expert, Active Directory Administrator	<input type="checkbox"/>
Hard Disk	<p>Check if hard disk usage is below 10% of available disk space on the drive.</p> <ul style="list-style-type: none"> On virtual machine: increase disk. On physical: mount bigger disk, reconfigure proxy to use the new partition. 	TFS Expert, Infrastructure Administration	<input type="checkbox"/>
Hardware, Other	<p>Proxy system freezes or acts slowly.</p> <p>If the proxy server is running on outdated hardware, check hard disk health and re-install it on new and more powerful hardware. See Hardware Recommendations¹⁵⁹ for the recommended hardware.</p>	TFS Expert, Infrastructure Administration	<input type="checkbox"/>

¹⁵⁹ <http://msdn.microsoft.com/en-us/library/vstudio/dd578644.aspx>

Planning Guide – Appendix

Area	Solution	Involved personas	Checked
Firewall, intrusion detection systems	Check: <ul style="list-style-type: none"> • If new firewall was installed or activated. • If Port 443 and/or 8081 are free to use. Not only on the proxy server itself, also on infrastructure around. • Use portqry to confirm if the port is open for connectivity to the TFS server. For example: • Portqry -n <tfsservername> -e 443 	TFS Expert, Security Administrator	<input type="checkbox"/>
Antivirus	Is there antivirus software installed on the proxy? Some antivirus tools block access to suspect files. Check the antivirus log file for recent virus activities.	TFS Expert, Security Administrator	<input type="checkbox"/>
Developers	Developers report receiving a proxy server error message in the Visual Studio output window that they cannot connect to the proxy server. Determine if the proxy server is accessible remotely via the proxy statistics URL.	TFS Expert, Infrastructure Admin	<input type="checkbox"/>

Table 50 – Walkthrough: Investigating Proxy failures

DR Recovery – Re-Install Team Foundation Proxy

Sometimes it's easier to install a new proxy server than it is to fix it, especially when there's a hardware failure or a software issues like a virus. Because the proxy just holds a copy of required files, you **don't lose** any data when you start over on a new system. See the following table for a setup quick start.

Step	Description	Reference Link(s)	Done
1	Verify the system requirements and server considerations.	See How to: Install Team Foundation Proxy and Set Up a Remote Site ¹⁶⁰ See Proxy Server Considerations , page 21.	<input type="checkbox"/>
2	Re-install Team Foundation Proxy.		<input type="checkbox"/>
3	Update the Team Foundation Proxy server to the same patch level as the Application tier.		<input type="checkbox"/>
4	Configure an isolated and manageable group of users who were impacted by the failure and verify that proxy failure symptoms are resolved.		<input type="checkbox"/>
5	Re-configure all users who were affected by the failure to use the new Team Foundation Proxy.		<input type="checkbox"/>

Table 51 - Walkthrough: Set up a new proxy server

Walkthrough: DR Recovery - Build Failure

SYMPTOMS

A group of developers has an issue with builds when code is checked in to TFS Build Server.

¹⁶⁰ <http://msdn.microsoft.com/en-us/library/vstudio/ee248710.aspx>

DR Investigation - Walkthrough

Area	Solution	Involved personas	Checked
End user Visual Studio	<p>Users reports that after checking in code from Visual Studio the build fails intermittently.</p> <ul style="list-style-type: none"> Check if build server is running. It could be down. If you can log in to the server, check if the TFS Admin Console accessible on the server. The log should be reviewed for changes that adversely affected the server. Check if the Build Controller is running, Check if the Build Agent is running, 	TFS Expert, End user	<input type="checkbox"/>
Service Account	<p>Confirm that the password still works. If it is disabled, enable it. If password has changed, reconfigure build and retype the password.</p>	TFS Expert, Active Directory Administrator	<input type="checkbox"/>
Enable Logging	<p>Enable logging on the build sever to get additional diagnostics information. Create a config file with the name "TFSBuildServiceHost.exe.config" and paste the following configuration information in it:</p> <pre><configuration> <system.diagnostics> <switches> <add name="BuildServiceTraceLevel" value="4"/> </switches> <trace autoflush="true" indentsize="4"> <listeners> <add name="myListener" type="Microsoft.TeamFoundation.TeamFoundationTextWriterTraceListener,M icrosoft.TeamFoundation.Common, Version=10.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" initializeData="C:\TFSBuildService Logs\TFSBuildServiceHost.exe.log" /> <remove name="Default" /> </listeners> </trace> </system.diagnostics> </configuration></pre> <p>For TFS 2012 Build Server, drop the file in this location (if you installed to the default location): C:\Program Files\Microsoft Team Foundation Server 11.0\Tools</p>	TFS Expert	<input type="checkbox"/>
Hard Disk	<p>Check if hard disk space is low.</p> <ul style="list-style-type: none"> On virtual machine: increase disk. On physical: mount bigger disk and reconfigure the build server to use the new partition. 	TFS Expert, Infrastructure Administration	<input type="checkbox"/>
Hardwar e, Other	<p>Build server system freezes or responds slowly.</p> <p>If the build server is running on outdated hardware, check hard disk health and re-install it on new and more powerful hardware.</p>	TFS Expert, Infrastructure Administration	<input type="checkbox"/>
Antivirus	<p>Is there antivirus software installed on the proxy? Some antivirus tools block access to suspect files. Check the antivirus log file for recent virus activities.</p>	TFS Expert, Security Administrator	<input type="checkbox"/>
Services	<p>Builds won't run. Check if Remote Connections Access Manager service is running. Company policies might not allow the Telephony service to run. This</p>	TFS Expert, Security Administrator,	<input type="checkbox"/>

Planning Guide – Appendix

Area	Solution	Involved personas	Checked
	is a problem because the RCMA service depends on Remote Connections Access Manager service.	Infrastructure Administration	

Table 52 - Walkthrough: Build failure investigation

DR Recovery – Re-Install Team Foundation Build Walkthrough

Build controller and agent on the same server

Step	Description	Reference Link(s)	Done
1.	Review the network topology.		<input type="checkbox"/>
2.	Confirm that there is a full backup of the TFS build server.		<input type="checkbox"/>
3.	Restore the Build server from the server backup.		<input type="checkbox"/>
4.	Re-established the connection to the TFS Application tier Server.		<input type="checkbox"/>
5.	Make sure that the build controller starts on the TFS build server.		<input type="checkbox"/>
6.	Build controller and agent on the same server.		<input type="checkbox"/>
7.	Make sure the build agent(s) starts on the TFS build server.		<input type="checkbox"/>
8.	Run the Best Practice Analyzer (BPA) on the TFS environment.		<input type="checkbox"/>
9.	If the server cannot be resorted from a backup then the OS has to be installed.		<input type="checkbox"/>
10.	Install the TFS build server software and configure it to connect to the TFS App Server.	How to: Install Team Foundation Build and Set Up ¹⁶¹	<input type="checkbox"/>
11.	Run the Best Practice Analyzer (BPA).	Best Practice Analyzer is part of the Power tool. To Install Power Tools ¹⁶²	<input type="checkbox"/>

Table 53 - Walkthrough: Build controller and agent on the same server

One TFS Build Controller and multi-Server TFS Build agents

Step	Description	Reference Link(s)	Done
1.	Review the network topology.		<input type="checkbox"/>
2.	If the TFS build controller failed: <ul style="list-style-type: none"> Confirm that there is a full backup of the TFS build controller server. Restore the TFS Build Controller server from the server backup. Re-established the connection to the TFS App Server. Re-established the connection to the TFS Build Agents server. Make sure the Build Controller starts on the TFS build server. Make sure that the Build agent(s) starts on the TFS build server. Run the BPA on the TFS environment. 	How to: Install Team Foundation Build and Set Up ¹⁶³ Best Practice Analyzer is part of the Power tool. To Install Power Tools ¹⁶⁴	<input type="checkbox"/>

¹⁶¹ <http://msdn.microsoft.com/en-US/library/vstudio/ms181712.aspx>

¹⁶² <http://visualstudiogallery.msdn.microsoft.com/b1ef7eb2-e084-4cb8-9bc7-06c3bad9148f>

¹⁶³ <http://msdn.microsoft.com/en-US/library/vstudio/ms181712.aspx>

¹⁶⁴ <http://visualstudiogallery.msdn.microsoft.com/b1ef7eb2-e084-4cb8-9bc7-06c3bad9148f>

Planning Guide – Appendix

Step	Description	Reference Link(s)	Done
3.	If the TFS build agents failed <ul style="list-style-type: none"> Confirm that there is a full backup of the TFS build agent server. Restore the failed TFS Build agent server from the server backup. Re-established the connection to the TFS Build controller server. Make sure that the Build agent(s) starts on the TFS build server. Run the BPA on the TFS environment. 	How to: Install Team Foundation Build and Set Up Best Practice Analyzer is part of the Power tool. To Install Power Tools	<input type="checkbox"/>

Table 54 - Walkthrough: One TFS Build Controller and multi-server TFS Build agents

Walkthrough: DR Recovery - SharePoint Failure

SYMPTOMS

A dedicated group of users—or all users—have issues with Team Portals; the impact could range from slow responses to an unreachable SharePoint Server. This can result in various errors.

NOTE

The attempt to upload a file that exceeds the maximum upload size will also result in an error or a timeout. This task isn't covered in this guide. For additional information, consult the disaster recovery strategy sources of SharePoint on [http://technet.microsoft.com/en-us/library/ff628971\(v=office.14\).aspx](http://technet.microsoft.com/en-us/library/ff628971(v=office.14).aspx) for SharePoint Server 2010 and <http://technet.microsoft.com/en-us/library/ff628971.aspx> for SharePoint Server 2013.

DR Investigation – Investigation Walkthrough

Area	Solution	Involved personas	Checked
Error Determination	Go to the SharePoint Central Administration and check for the first error(s).	TFS / SharePoint Expert	<input type="checkbox"/>
Security	Question to ask... <ul style="list-style-type: none"> Do all reported symptoms belong to a team or a special group like "testers"? Check the security settings for this group and reconfigure it if needed.	TFS / SharePoint Expert Administrator (Project Administrator)	<input type="checkbox"/>
SharePoint Service Account	Confirm that the password still works and that the SharePoint service account is not disabled. If it is disabled, enable it. If the password has changed, reconfigure SharePoint and retype the password.	TFS / SharePoint Expert Active Directory Administrator	<input type="checkbox"/>
Database Connection and SQL Server	Connect to the SharePoint database via Microsoft SQL Server Management Console, and check the SQL server health: CPU, memory, and disk usage.	TFS Expert, SQL Server Administrator	<input type="checkbox"/>
Firewall, intrusion detection systems	Check: <ul style="list-style-type: none"> If a new firewall was installed or activated. Some necessary ports are 1434 TCP and 2383 UDP, 137,138,139, 17012 TCP. Not only on the SharePoint server itself, also on infrastructure around. Plan security hardening for SharePoint 2013¹⁶⁵ for more information on ports and security hardening of SharePoint. 	TFS Expert, Network Administrator	<input type="checkbox"/>

¹⁶⁵ <http://technet.microsoft.com/en-us/library/cc262849.aspx>

Planning Guide – Appendix

Area	Solution	Involved personas	Checked
	If applicable, shut down the firewall or switch to a lower security level to easily diagnose this problem.		
Virus	Check your antivirus solution. Some viruses can cause errors on http communications.	TFS Expert, Security Administrator	<input type="checkbox"/>
SharePoint Logging	Review the logs in C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\12\LOGS to see if errors are being captured.	TFS/SharePoint Expert	<input type="checkbox"/>
IIS Logging for SharePoint	C:\Inetput\Logs\... <location of SharePoint IIS logging>. Review connections being made/not made by users. Do you see any status connections other than 200s? Are users connecting?	TFS/SharePoint Expert, Network Administrator	<input type="checkbox"/>
SharePoint Content Database	Determine if data is being logged into the SharePoint database for the TPC sites <pre>SELECT TOP 100 * FROM [WSS_Content_<DB>] . [dbo] . [EventLog] ORDER BY EventTime DESC</pre>	SharePoint Expert	<input type="checkbox"/>

Table 55 - Walkthrough: One TFS Build Controller and multi-server TFS Build agents

Authoring Reports

There are many sources for report information with TFS. You can use the Perfmon counters, data captured in the TFS databases, SQL Server statistics, the TFS API, and even the Windows event logs. Using these sources gives you a fairly complete picture of the state of your TFS environment.

Authoring Reports Guidance Walkthroughs

These walkthroughs are intended as guidance for creating reports and also to help you identify metrics that point to the source of the smoke before it becomes a fire.

Reporting With TFS Data

There is a quite a bit of information within the TFS databases. You can use pre-established reports like Grant Holliday's [Administrative Report Pack](#)¹⁶⁶ or you can develop your own. Our recommendation is that if you develop your own, you should involve a database administrator who can review the queries you create so you don't adversely affect the TFS environment.

Step	Description	Details	Done
1	Install the Administrative Report Pack.	Download and install Grant's reports. Depending on your environment configuration, you might need DBA assistance. The TFS implementation could have restrictions on who can perform particular Reporting Service activities such as configuring data sources.	<input type="checkbox"/>
2	Review the Reports.	Review the information available in the admin reports. Determine which ones should be emailed to the TFS support team members on a regularly scheduled basis.	<input type="checkbox"/>
3	Understand how to use reporting tools like Report Builder.	See Getting Started with Report Builder 3.0 ¹⁶⁷ for more details.	<input type="checkbox"/>
4	Review how to create customized reports.	See Create, Customize, and Manage Reports for Visual Studio ALM ¹⁶⁸ and Creating and Customizing Reports for TFS ¹⁶⁹ more details.	<input type="checkbox"/>
5	Leverage existing views.	The TFS Warehouse has view options to report on activity. Use these views as a basis for your reports and implement proper joins.	<input type="checkbox"/>

Table 56 – Walkthrough: authoring reports

Reporting with SQL Server

Using SQL Server views to see how performance is in SQL Server.

Step	Description	Details	Done
1	Use SQL Server Management Server to query data.	Sys Admin access is required in SQL Server.	<input type="checkbox"/>
2	Identify what you need.	Use Activity Monitor to see performance information. Activity Monitor is an application built into SQL Server for monitoring.	<input type="checkbox"/>

¹⁶⁶ <http://blogs.msdn.com/b/granth/archive/2010/07/12/administrative-report-pack-for-team-foundation-server-2010.aspx>

¹⁶⁷ <http://technet.microsoft.com/en-us/library/dd220460.aspx>

¹⁶⁸ <http://msdn.microsoft.com/en-us/library/vstudio/dd578644.aspx>

¹⁶⁹ <http://msdn.microsoft.com/en-us/library/ff647430.aspx>

Planning Guide – Appendix

Step	Description	Details	Done
3	Identify if reporting service reports are running long.	Use the ExecuteLog2 view to view long running reports. Reports that capture a lot of data can slow down other reports that are currently running.	<input type="checkbox"/>

Table 57 – Walkthrough: Reporting with SQL Server

Reporting with Log Parser

Log Parser provides a SQL like language for selecting information from event or IIS logs. It is a free tool and plenty of examples can be found on how it is used to extract data to be analyzed.

Step	Description	Details	Done
1	Install Log Parser.	Download and install Log Parser 2.2.	<input type="checkbox"/>
2	Identify what you need.	logparser.exe -i:EVT "SELECT * FROM Application WHERE TimeGenerated > '2013-01-1 00:00:00' AND EventType IN (1;2) ORDER BY TimeGenerated DESC" -o:CSV -q:ON -stats:OFF >> c:\temp\TFSAppEvents.csv	<input type="checkbox"/>
3	Review the errors that are shown.	Analyze the results. <ul style="list-style-type: none"> Are there patterns? Are users who raising issues being shown in the error log? Sample errors: “Unable to connect to database. Check database connection information and make sure the database server is running.” “Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding.”	<input type="checkbox"/>
		See Download Log Parser 2.2 ¹⁷⁰ for more details	<input type="checkbox"/>

Table 58 – Walkthrough: Reporting with Log Parser

Reporting With Perfmon

The PAL tool is primarily a PowerShell script that analyzes performance monitor logs by using arguments and parameters that are passed to it to.

Step	Description	Details	Done
1	Install PAL.	Download PAL from CodePlex.	<input type="checkbox"/>
2	Review PAL functionality.	Review how data can be selected with PAL.	<input type="checkbox"/>
3	Determine the relevant information to review.	Analyze the results of importing performance counter files. <ul style="list-style-type: none"> What is the user patterns like? Items to monitor <ul style="list-style-type: none"> Items flagged on the report 	<input type="checkbox"/>
		See PAL CodePlex download ¹⁷¹ for more details	<input type="checkbox"/>

Table 59 – Walkthrough: Reporting with Perfmon

¹⁷⁰ <http://msdn.microsoft.com/en-us/library/vstudio/dd578644.aspx>

¹⁷¹ <http://msdn.microsoft.com/en-us/library/vstudio/dd578644.aspx>

Walkthrough-Create Report to Monitor Average Response Time Counters

Step1: Create Counters to monitor

To collect the counters to monitor, we will use Logman. Logman creates and manages Event Trace Session and Performance logs and supports many functions of Performance Monitor from the command line.

The first step is to create a text file (c:\TFSCouters.txt) with all the counters you want to monitor. For the current scenario, we will be monitoring the Average Response Time Counter.

NOTE

\TFS Services\Average Response Time

Step 2: Collect Data

To begin collection of the performance counter, run the following command on the TFS Server to create Performance Counter Collection.

```
logman create counter TFSCounterCollection -s %computername% -cf "c:\TFSCouters.txt"
```

After the collection is created, run the following command to begin the counter collection:

```
logman TFSCounterCollection Start
```

To end the Collection, run this command:

```
logman TFSCounterCollection Stop
```

By default, on Windows 2008 R2 servers, the performance monitor counters will be stored in %systemdrive%\PerfLogs\Admin and will be named after the collection name (in the previous case, they will be called TFSCounterCollection.blg).

Step 2: Analyze Data

After the performance logs are created, we can either move the counters to SQL database for analysis or export the logs to .CSV format and use the reporting capabilities of Excel to analyze the data.

In this example, we will export the performance counters to .CSV format and use reporting capabilities of Excel to analyze the data.

To export the counter data to a CSV, run the following command from the directory where the file containing the counters (TFSCounterCollection.blg) are stored:

```
Relog TFSCounterCollection.blg -f csv -o TFSLogFile.csv
```

Opening the CSV file in Excel lets you to use the Pivot Charts functionality in Excel to analyze the results. The image below shows how data can be viewed in Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1																							
2	5/17/12 7:26 PM	0	0	0	0	0	0	0	0	0	34,980,016,128	81,268,885	689,602	15	0	6	0	84,633,165	0	6			
3	5/17/12 7:26 PM	0	0	0	0	0	0	0	0	0	34,980,016,128	81,268,885	689,602	15	0	6	0	15 84,633,384	0	6	15		
4	5/17/12 7:27 PM	0	0	0	0	0	0	0	0	0	34,980,098,048	81,269,273	689,604	13	331	7	0	13 84,633,575	0	6	13		
5	5/17/12 7:27 PM	0	0	0	0	0	0	0	0	0	34,980,098,048	81,269,414	689,604	10	0	5	0	9 84,633,716	0	5	10		
6	5/17/12 7:27 PM	0	0	0	0	0	0	0	0	0	34,980,098,048	81,269,532	689,604	8	0	7	0	7 84,633,824	0	7	9		
7	5/17/12 7:27 PM	0	0	0	0	0	0	0	0	0	34,980,134,912	81,269,595	689,608	6	1	5	0	5 84,633,901	0	5	7		
8	5/17/12 7:28 PM	0	0	0	0	0	0	0	0	0	34,980,134,912	81,269,595	689,608	24	0	5	0	24 84,634,255	0	6	25		
9	5/17/12 7:28 PM	0	0	0	0	0	0	0	0	0	34,980,134,912	81,270,150	689,608	16	0	5	0	14 84,634,466	0	5	18		
10	5/17/12 7:28 PM	0	0	0	0	0	0	0	0	0	34,980,134,912	81,270,376	689,608	15	0	5	0	14 84,634,682	0	5	16		
11	5/17/12 7:28 PM	0	0	0	0	0	0	0	0	0	34,980,327,424	81,270,585	689,610	15	0	5	0	14 84,634,893	0	5	16		
12	5/17/12 7:29 PM	0	0	0	0	0	0	0	0	0	34,980,327,424	81,270,783	689,610	14	0	5	0	13 84,635,091	0	5	14		

Figure 47 - Excel Pivot Charts example

Sample Maintenance Scripts

Detecting circular Active Directory relationships which can degrade Performance

Overview

The script:

- Prints out hierarchy height in Active Directory relationships.
- Detects circular AD relationships.

Sample Script

```

=====
' VBScript Source File -- Created with SAPIEN Technologies PrimalScript 4.0
'
' NAME: GrpNestingWeights.vbs
'
' AUTHOR: Guy Teverovsky, Microsoft
' DATE   : 3/23/2009
'
' COMMENT: The script is useful for identifying "heavy" groups - those that
'           when added to user will "donate" a lot of groups to user token
'           through group nesting. The script will output the list of all
'           security enabled groups in the domain and will tell how many
'           groups each one "donates".
'
'           i.e.:      if GRP1 is member of GRP2 and GRP3 and GRP3 is member
'                       of GRP4, then membership in GRP1 "donates" 4 groups
'
'           This will not help you to calculate user access token, but will
'           give you a good estimate of over-abused groups you want to
'           take care of.
'
'           P.S.:      As a bonus, the script will also detect nesting loops -
'                       if GRP1 is member of GRP2 and GRP2 is member of GRP3,
'                       and GRP3 is member of GRP1, you will be notified about
'                       the loop.
'
' COPYRIGHT:
'           Copyright © Microsoft Corporation. All Rights Reserved. This code
'           is released under the terms of the Microsoft Public License (MS-PL,
'           http://opensource.org/licenses/ms-pl.html.)
'           This is sample code only, do not use in production environments!
=====

```

Option Explicit

Const LOG_NAME = "NestedGroupsLog.txt"

Const ADS_GROUP_TYPE_SECURITY_ENABLED = &H80000000

```

' File actions
Const ForReading      = 1
Const ForWriting      = 2
Const ForAppending    = 3
'
Const TristateUseDefault = -2
Const TristateTrue      = -1
Const TristateFalse     = 0

```

Dim objRS, objConn, objAdoCmd

```

Dim objRootDSE
Dim objGroupList
Dim strBaseDn
Dim iGroupCount : iGroupCount = 0

```



```

Dim objFSO
Dim objLogFile

Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objLogFile = objFSO.CreateTextFile(LOG_NAME, ForWriting, TristateTrue)

Set objConn = CreateObject("ADODB.Connection")
objConn.Provider = "ADSDSOObject"
objConn.Open "Ads Provider"
Set objAdoCmd = CreateObject("ADODB.Command")
objAdoCmd.ActiveConnection = objConn
objAdoCmd.Properties("Page Size") = 1000

Set objGroupList = CreateObject("Scripting.Dictionary")
objGroupList.CompareMode = vbTextCompare

Set objRootDSE = GetObject("LDAP://RootDSE")
'strBaseDn = BASE_RDN & "," & objRootDSE.Get("DefaultNamingContext")
strBaseDn = objRootDSE.Get("DefaultNamingContext")

EnumGroups

objLogFile.Close

Set objLogFile      = Nothing
Set objRootDSE      = Nothing
Set objGroupList    = Nothing
Set objRS           = Nothing
Set objConn         = Nothing

' Loop through all the security enabled groups
Function EnumGroups
    Dim strLdapSearchString
    Dim iGroupWeight

    strLdapSearchString = "<LDAP://" & strBaseDn & _
">;(&(objectCategory=group)(groupType:1.2.840.113556.1.4.803:=2147483648));adspath,distinguishedName,sAMAccountName;subtree"

    objAdoCmd.CommandText = strLdapSearchString

    Set objRS = objAdoCmd.Execute(strLdapSearchString)

    While Not objRS.EOF
        objGroupList.RemoveAll
        iGroupCount = iGroupCount + 1

        iGroupWeight = GetGroupNestingWeight(objRS.Fields(0).Value)
        OutputLine objRS.Fields(2).Value & vbTab & iGroupWeight, False

        OutputLine objRS.Fields(2).Value & vbTab & iGroupWeight & _
vbTab & objRS.Fields(1).Value, True
        objRS.MoveNext
    Wend
End Function

Function GetGroupNestingWeight(strGroupAdsPath)
    Dim objGroup
    Dim objMemberOf, strMemberDn
    Dim iGroupWeight

    iGroupWeight = 1
    GetGroupNestingWeight = 1

    Set objGroup = GetObject(strGroupAdsPath)

    On Error Resume Next
        TypeName(objGroup.GetEx("memberof"))
        If Err.Number <> 0 Then Exit Function
    On Error GoTo 0

```

```
For Each strMemberDn In objGroup.GetEx("memberof")
    strMemberDn = Replace(strMemberDn, "/", "\/")
    Set objMemberOf = GetObject("LDAP://" & strMemberDn)

    If LCase(objMemberOf.class) = "group" Then
        'Loop detection safety
        If LCase(objRS.Fields(2).Value) =
            LCase(objMemberOf.sAMAccountName) Then
            OutputLine "!!! LOOP Detected !!!!" & vbTab & _
                objRS.Fields(2).Value & vbTab & "==" & _
                objGroup.sAMAccountName, True

            OutputLine "!!! LOOP Detected !!!!" & vbTab & _
                objRS.Fields(2).Value & vbTab & "==" & _
                objGroup.sAMAccountName, False
            'We have not visited the group yet
        ElseIf Not objGroupList.Exists(objMemberOf.sAMAccountName) Then
            objGroupList.Add objMemberOf.sAMAccountName,
                objMemberOf.AdsPath
            'Count only security groups
            If (objMemberOf.groupType And
                ADS_GROUP_TYPE_SECURITY_ENABLED) <> 0
            Then
                iGroupWeight = iGroupWeight +
                    GetGroupNestingWeight(objMemberOf.AdsPath)
            End If
        End If
    End If
Next
GetGroupNestingWeight = iGroupWeight
End Function

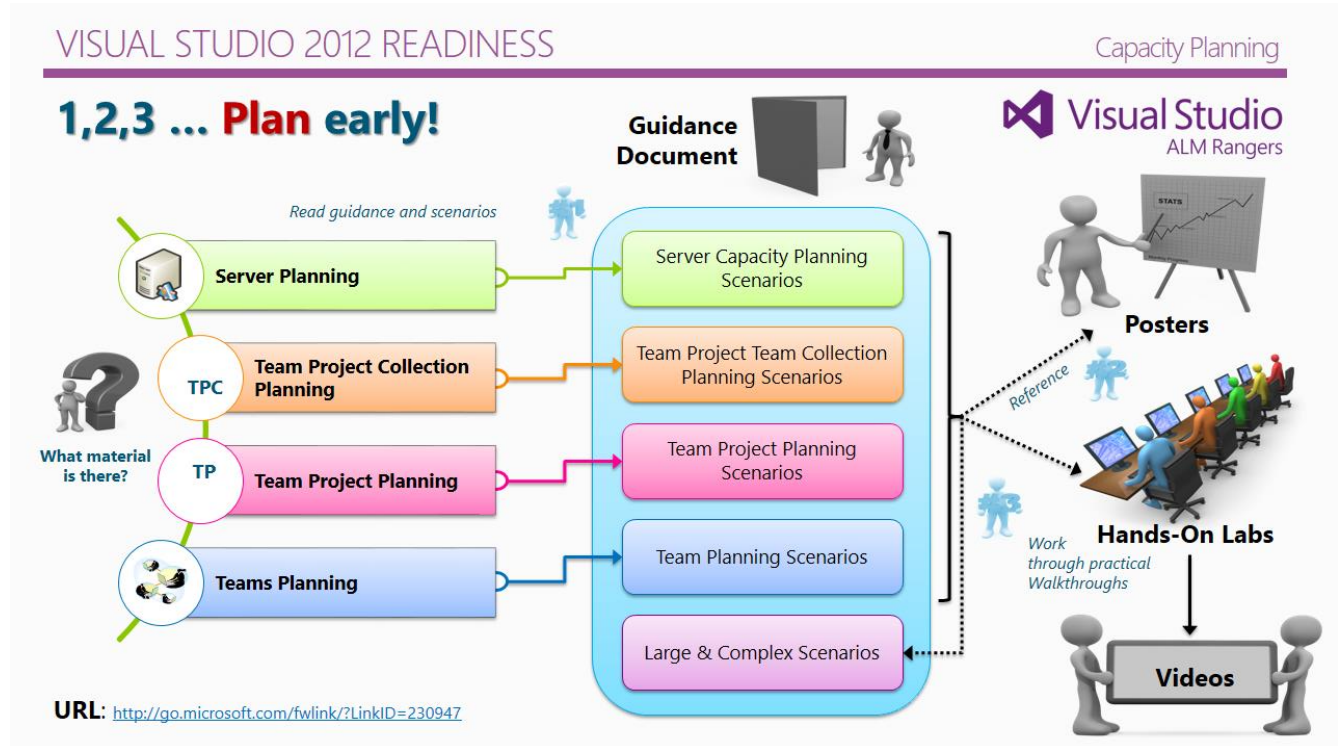
Function OutputLine(strText, boolLogToFile)
    If Not boolLogToFile Then WScript.Stdout.WriteLine strText
    'WScript.Stderr.WriteLine strText
    If boolLogToFile Then objLogFile.WriteLine strText
End Function
```

Constraints

Note that the script:

- Is **not** supposed to serve as an example of good programming practices.
- Does **not** account for cross-domain loops.

Quick Reference Sheets



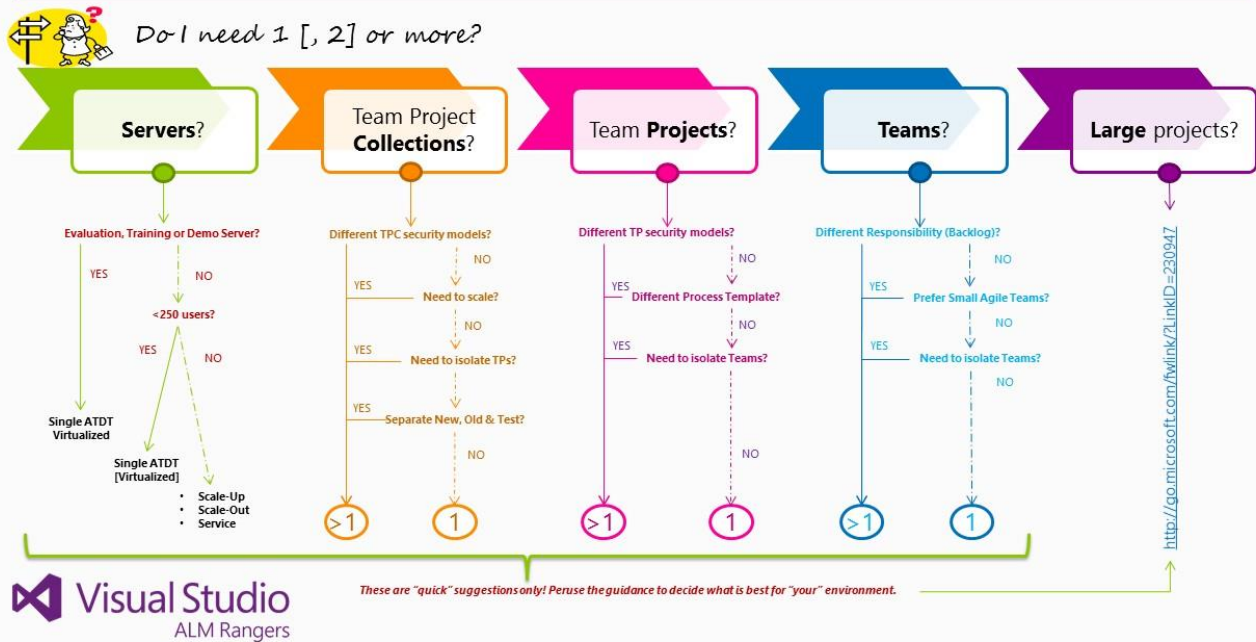
Other quick reference posters featured in this section:

- Server Planning Overview
- Server Planning
- Team project Collection Planning
- Team Project Planning
- Teams Planning
- Disaster Recovery Planning

High **quality** and **full-page** quick reference poster versions are available in the guidance downloads.

VISUAL STUDIO 2013 READINESS

Server Planning



VISUAL STUDIO 2013 READINESS

Server Planning

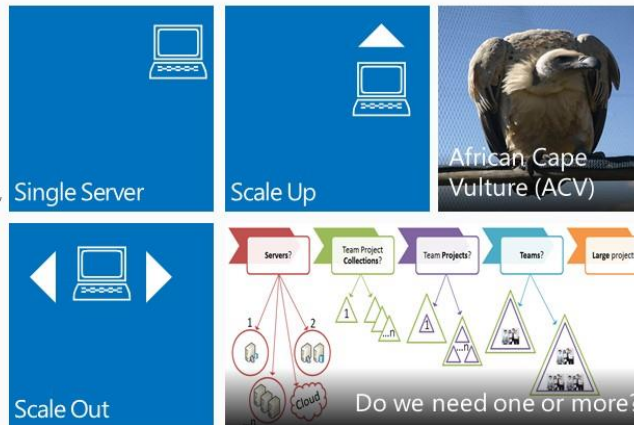
Overview

The Team Foundation Server Capacity Planning introduces common planning steps for hosted and on-premises deployment models such as:

- Single Server
- Multi Server (scale up)
- Multi Server (scale out)

Key planning decisions must be considered, such as:

- Hosted versus on-premises
- Number of users
- Number of Team Project Collections
- Number of Team Projects
- Virtualization



¹ Ref: Refers to the typical on-premises hardware specification, as covered in the guidance documentation.

References

- Visual Studio ALM Rangers <http://msdn.microsoft.com/en-us/vstudio/ee358786.aspx>
- Ranger Solutions <http://msdn.microsoft.com/en-us/vstudio/ee358787>



You can plan an on-premises infrastructure based on maximum users, active team project collections and/or team projects

Max Users Planning

Ref	Max Users	Deployment Model
1	250	<input type="checkbox"/>
2	500	<input type="checkbox"/>
3	2200	Multi Server (scale up) <input type="checkbox"/>
4,5,6	3600	<input type="checkbox"/>
7	3600+	Multi Server (scale out) <input type="checkbox"/>

Team Project Collection Planning

Ref ¹	Max Users	Active TPC
1	250	5 <input type="checkbox"/>
2	500	10 <input type="checkbox"/>
3	2200	75 <input type="checkbox"/>
4		90 <input type="checkbox"/>
5	3600	125 <input type="checkbox"/>
6		195 <input type="checkbox"/>
7	3600+	75 / SQL Server Instance <input type="checkbox"/>

Team Project Process Template Planning

TPCs =	[] Agile TPs / 1000
	+ [] Scrum TPs / 1000
	+ [] CMMI TPs / 250
3	<input type="checkbox"/>

Virtualized Environment

Rule of thumb is to reduce the maximum users by 20% or increase the hardware specifications when considering a virtualized environment.

1	Max Users	Max Users - 20% <input type="checkbox"/>
2	Hardware Specs	Typical + 20% <input type="checkbox"/>

VISUAL STUDIO 2013 READINESS

Team Project Collection Planning

Overview

Team Foundation Server 2010 introduced the feature of Team Project Collections.

Each Team Foundation Server can host one or more Team Project Collections, each of which can accommodate one or more Team Projects.

Similar to SharePoint Collections, the Team Foundation Server Team Project Collection acts as a container and as a basic unit of isolation and archiving.

Team Project Collections support:

- Scalability
- Backup and Restore
- Recovery
- Security isolation



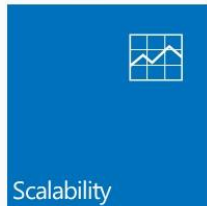
Backup & Restore



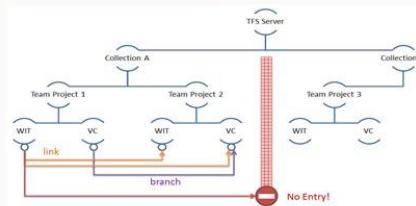
Security isolation



African Cape Vulture (ACV)



Scalability



Understand the Team Project Collections (TPCs) by exploring the isolation considerations, the constraints and the planning considerations.

Planning Considerations

1	100 active TPCs per server	<input type="checkbox"/>
2	8000 total TPCs per SQL Server instance	<input type="checkbox"/>

Isolation Considerations

1	Scalability	<input type="checkbox"/>
2	Backup and Restore	<input type="checkbox"/>
3	Recovery	<input type="checkbox"/>
4	Security isolation	<input type="checkbox"/>

Constraints

1	No branching of version control across TPCs	<input type="checkbox"/>
2	No sharing of WIs across TPCs	<input type="checkbox"/>
3	No sharing of queries across TPCs	<input type="checkbox"/>
4	No sharing of reports across TPCs	<input type="checkbox"/>
5	No renaming of TPC name	<input type="checkbox"/>
6	Build machines scoped to one TPC	<input type="checkbox"/>
7	Connect to one TPC at a time	<input type="checkbox"/>



References

- Visual Studio ALM Rangers <http://msdn.microsoft.com/en-us/vstudio/ee358786.aspx>
- Ranger Solutions <http://msdn.microsoft.com/en-us/vstudio/ee358787>
- Organizing your Server with Team Foundation Collections <http://msdn.microsoft.com/en-us/magazine/gg983486.aspx>

VISUAL STUDIO 2013 READINESS

Team Project Planning

Overview

Imagine a team project as some kind of container for related data within a defined scope. Even though the term "Team Project" implies that the scope of a team should somehow relate to a team of developers – that's not necessarily the case.

A team project consists of related data. Most the data of a team project is stored within one single team project collection and therefore in a single database, with other data (like documents and reports) stored in SharePoint and Reporting Services. The main purpose of working with multiple Team Projects within a single Team Project Collection is rather a logical isolation of data than a physical isolation.

Team Project Collections support:

- Security isolation
- Process template isolation



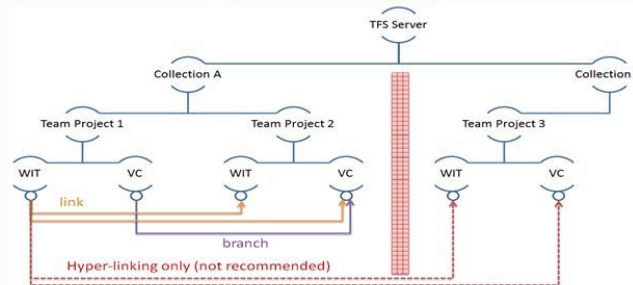
Process Template Isolation



Security Isolation



African Cape Vulture (ACV)



Understand the Team Projects (TPs), by exploring the isolation considerations, the constraints and the planning considerations.

Planning Considerations

1	1000 Agile based TPs per TPC	<input type="checkbox"/>
2	1000 Scrum based TPs per TPC	<input type="checkbox"/>
3	250 CMMI based TPs per TPC	<input type="checkbox"/>

Isolation Considerations

1	Security	<input type="checkbox"/>
2	Process template	<input type="checkbox"/>

Constraints

1	No sharing of areas & iterations across Team Projects	<input type="checkbox"/>
2	No sharing of build definitions across TPs	<input type="checkbox"/>
3	No sharing of WI Categories across TPs	<input type="checkbox"/>
4	No sharing of WI Templates across TPs	<input type="checkbox"/>
5	No sharing of WIT Definitions across TPs	<input type="checkbox"/>
6	Team Project name is immutable	<input type="checkbox"/>



References

- Visual Studio ALM Rangers <http://msdn.microsoft.com/en-us/vstudio/ee358786.aspx>
- Ranger Solutions <http://msdn.microsoft.com/en-us/vstudio/ee358787>
- Organizing your Server with Team Foundation Collections <http://msdn.microsoft.com/en-us/magazine/gg983486.aspx>

VISUAL STUDIO 2013 READINESS

Teams Planning

Overview

Team Foundation Server Teams are new and they are defined within a Team Foundation Server Team Project.

The Team Foundation Server Teams concept grew out of Scrum Team needs.

- Team Foundation Server Teams allow you to set up a team that has its own Product Backlog to manage from the Team Foundation Server Team Project's Product Backlog.
- Team Foundation Server Teams manage their own Area Paths in a Team Foundation Server Project and become a collaboration hub for the team.
- The Team Foundation Server Team ends up being a named group of team members who work on a demarcated area of the Team Project.



Own Product Backlog



Own Area Paths



African Cape Vulture (ACV)

Understand the Teams by exploring the isolation considerations, the constraints and the planning considerations.

Planning Considerations

- 1 Teams cannot be shared across Team Projects ☐
- 2 Teams are flat user lists ☐

Team Advantages

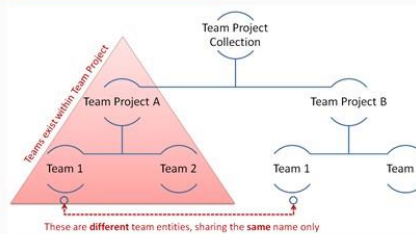
- 1 Categorize large teams into sub-teams ☐
- 2 Allocate separate backlogs to teams ☐

Constraints

- 1 Agile teams are typically <10 ☐
- 2 Groups >100 not loaded by Team Explorer ☐



Demarcated Area



References

- Visual Studio ALM Rangers <http://msdn.microsoft.com/en-us/vstudio/ee358786.aspx>
- Ranger Solutions <http://msdn.microsoft.com/en-us/vstudio/ee358787>
- Configure a master backlog and sub-teams <http://tinyurl.com/Backlog-Teams>



VISUAL STUDIO 2013 READINESS

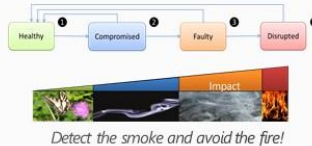
Disaster Recovery Planning

Overview

Disaster recovery (DR) is the process needed to recover infrastructures and services after being disrupted due to a man-made and natural disaster. Both are very difficult to predict and prevent, making it important for organizations to define, implement and continuously evaluate processes, procedures and policies to mitigate the risk for critical infrastructures and services.

Disaster recovery planning typically encompasses **preventive**, **corrective** and **detective** measures.

The ALM Rangers Planning Guide contains guidance for an **avoidance strategy** to proactively avoidable disaster recovery caused by infrastructure and solution degradation, as well as practical walkthroughs to help you through the **recovery process** if needed.



Planning



Avoidance



African Cape Vulture (ACV)

Guidance

You can find the following guidance in the planning guide, to assist you with typical disaster recovery. Planning and monitoring:

- Understanding DR Strategies
- Understanding DR Avoidance
- Active monitoring and reporting

Walkthroughs

You can find the following walkthroughs in the planning guide, to assist you with typical disaster recovery:

- Planning
- Complete failure (fire, natural disaster)
- Data tier (DT) failure
- Application Tier (AT) failure
- Proxy failure
- Build failure
- SharePoint failure
- Switch-over to secondary site



Recovery



References

- Visual Studio ALM Rangers <http://msdn.microsoft.com/en-us/vstudio/ee358786.aspx>
- Ranger Solutions <http://msdn.microsoft.com/en-us/vstudio/ee358787>
- Planning Guide <http://aka.ms/treasure5>

