LINK Data Dispatcher



ETFS Governance

Prepared for

3M

3/6/2015

Version 1.1 Final

Prepared by

**Tony Feissle**

Senior Consultant

[tfeissle@micrososoft.com](mailto:tfeissle@micrososoft.com)

Contributors

**Louis Weinstein**

|  |  |
| --- | --- |
|  | Prepared for 3M |

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change reference |
| 2/10/2015 | Tony Feissle | 1.0 | Draft for review/discussion |
| 3/6/2015 | Tony Feissle | 1.1 | Reflects Michael and groups comments |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version approved | Position | Date |
| Louis Weinstein | 1.0 | Principal Consultant | 2/16/2015 |
| Michael O’Brien |  | Software Engineer Specialist |  |
|  |  |  |  |

Table of Contents

[1 3M TFS Governance 4](#_Toc413397600)

[Introduction 4](#_Toc413397601)

[Definitions and Acronyms 5](#_Toc413397602)

[1.1 Diagram Symbols 6](#_Toc413397603)

[2 Business Goals 7](#_Toc413397604)

[2.1 3M Environment 7](#_Toc413397605)

[2.2 Current Implementation Issues and Risks 7](#_Toc413397606)

[3 Shared Services and Security Governance 8](#_Toc413397607)

[3.1 Team Boundaries and Control 8](#_Toc413397608)

[3.2 Shared Services 9](#_Toc413397609)

[3.2.1 TFS SQL Server 9](#_Toc413397610)

[3.2.2 Team Project Collections 9](#_Toc413397611)

[3.2.3 Team Projects 10](#_Toc413397612)

[3.3 Security 10](#_Toc413397613)

[3.3.1 Active Directory Users and Groups 10](#_Toc413397614)

[3.3.2 TFS Permissions 10](#_Toc413397615)

[3.3.2.1 TFS Permissions for Scrum projects 12](#_Toc413397616)

[3.3.3 Source Control Security 12](#_Toc413397617)

[3.3.4 Microsoft Office SharePoint® Server (SharePoint) 13](#_Toc413397618)

[3.3.5 SQL Server Reporting Services 13](#_Toc413397619)

[3.3.6 SQL Server Analysis Services 13](#_Toc413397620)

[4 Process Governance 14](#_Toc413397621)

[4.1 Onboarding and Migrating 14](#_Toc413397622)

[4.2 Branching and Merging Models 15](#_Toc413397623)

[4.2.1 Default Branching and Merging Option 15](#_Toc413397624)

[4.2.2 Dual Project Development Branching and Merging Option 16](#_Toc413397625)

[4.2.3 Parallel Project Teams Branching and Merging Option 16](#_Toc413397626)

[4.3 Tool Integration 17](#_Toc413397627)

[4.4 TFS Updates and Upgrades 17](#_Toc413397628)

[4.5 TFS Process Templates 17](#_Toc413397629)

[4.6 Checkin Policies 20](#_Toc413397630)

[4.7 Change Board 21](#_Toc413397631)

[5 Build Service Governance 22](#_Toc413397632)

[5.1 Continuous Integration Build Servers 22](#_Toc413397633)

[5.2 Daily Build Servers 22](#_Toc413397634)

[5.3 Build Definition 22](#_Toc413397635)

[5.4 Symbol Server 22](#_Toc413397636)

[5.5 Release Management 23](#_Toc413397637)

[6 Integration Governance 24](#_Toc413397638)

[6.1 Third Party Tools or Home Grown Tools 24](#_Toc413397639)

[7 Support Governance 25](#_Toc413397640)

[7.1 Team Foundation Server Support 25](#_Toc413397641)

[7.2 Customization Policies 25](#_Toc413397642)

[7.3 Service Level Agreements 25](#_Toc413397643)

[7.4 Team Foundation Server Monitoring 26](#_Toc413397644)

[7.5 Governance and ETFS Training 27](#_Toc413397645)

[8 Executive Summary 28](#_Toc413397646)

[9 Next Steps 29](#_Toc413397647)

[10 Resources and References 30](#_Toc413397648)

[10.1 Shared Services and Security 30](#_Toc413397649)

[10.2 Processes 30](#_Toc413397650)

[10.3 Build Services 31](#_Toc413397651)

[10.4 Integration 31](#_Toc413397652)

[10.5 Support 32](#_Toc413397653)

[ALM Assessments 33](#_Toc413397654)

[Architecture & Design 33](#_Toc413397655)

[Software Configuration Management 34](#_Toc413397656)

[Governance 35](#_Toc413397657)

[Deployment & Operations 36](#_Toc413397658)

[Testing & Quality Assurance 37](#_Toc413397659)

[Project Planning & Management 38](#_Toc413397660)

[Development 39](#_Toc413397661)

[Requirement Engineering & UX 40](#_Toc413397662)

1. 3M TFS Governance

## Introduction

The Team Foundation Server (TFS) environment should be governed in a similar fashion to other applications on the 3M network. In this document, Microsoft presents recommendations for the governance plan to be leveraged by the Enterprise Team Foundation Servers at 3M. The governance recommendations are based on an understanding of 3M’s current organization model and desired TFS environmental goals. There are five areas of governance that are covered; Security, TFS Processes, Build Services, TFS Integration and Support. Overall the TFS governance plan is to help 3M manage team projects, Enterprise TFS maintenance and minimize unplanned work for the Enterprise TFS.

At the beginning of the TFS assessment at 3M, several groups (HIS & IPD) graciously filled out the ALM Rangers Application Lifecycle Management Assessment. This survey is a self-assessment tool which is designed to surface both ALM techniques and technologies which can be improved. While this document will review some of the more pertinent sections of this assessment, the purpose is not meant to be “permanent grade”, but rather a baseline by which a team can measure areas which need improvement and aid in the creation of an action plan. The results are discussed in the following section: [ALM Assessment](#_HIS_Assessment).

There are many factors that 3M needs to consider in determining how to support and maintain the Enterprise Team Foundation Server (ETFS) environment:

* Application or project team size
* Complexity of application or project team
* Number of applications or project teams in the organization
* Business or regulatory requirements
* IT infrastructure performance

This document using the above considerations will provide recommendations and guidance for governance of the Enterprise Team Foundation Servers.

Definitions and Acronyms

**ABT** – Application Build Team

**AD** – Active Directory

**ALM** – Application Lifecycle Management

**BVT** - Build Verification Test

**CI** – Continuous Integration

**CMMI** – Capability Maturity Model Index

**DevOps** – Development Operations

**DR** – Disaster Recovery

**DTA** – Development Tools Administration

**EALMS** – Enterprise Application Lifecycle Management System

**ETFS** – Enterprise Team Foundation Server

**IP** – Intellectual property

**MMC** – Microsoft Management Console

**TFS** – Team Foundation Server

**TSC** – Tools Support Center

**TPC** – Team Project Collection

**TP** – Team Project

**SLA** – Service Level Agreement

**SSAS** – SQL Server Analysis Services

**SSRS** – SQL Server Reporting Services

**UAT** – User Acceptance Testing

* 1. Diagram Symbols

In this document the branching and merging diagrams use the following Symbols

* B – Branch
* BM – Baseless Merge
* FI – Forward Integration
* RI – Reverse Integration
* 🔒 indicates read-only
* 🗶 indicates deleted branch

1. Business Goals

3M is interested in defining and setting the standards and governance for supporting and managing application teams via an Enterprise Team Foundation Server system. These standards will define the areas that the DTA team should manage within their scope. 3M is also looking to reduce the number of exceptions that the application and project teams request when being audited. The business needs to have a centralized solution for managing its Intellectual Property (IP) assets, and provide tools to help developers improve the quality of the applications.

* 1. 3M Environment

3M currently has several project teams/divisions using different templates as part of the ETFS system. Because of the business requirements for managing application quality, 3M needs to provide an Enterprise Team Foundation Server environment that follows standards for all project teams.

Today 3M cannot enforce consistent business requirements or processes for software development across all of the project teams. This is important to 3M in order to ensure quality applications are built and tested and to reduce the number of variances being requested.

The environment at 3M has a TFS clustered application tier with one or more Team Project Collections, and a proposed virtualized on-demand build servers to serve the needs for all team projects. Most build servers are currently running all of the time (even when **not** needed).

* 1. Current Implementation Issues and Risks

The current issues and risks for 3M are:

* Inconsistent software development processes and source control
* Resources not able to move between teams without learning a new process
* Inconsistent branching and merging models
* Multiple tools or processes used within project teams
* The ALM security for the standalone environments is not enforced or standardized
* No Executive level reporting
* No centralized build servers or services
* No centralized logs or data for audits
* No auditing for tools or open source code licensing
* Unknown number of development efforts
* No traceability
* Auditors are not sure that all applications have been reviewed since it is not centralized
* No source code reuse or tools sharing for doing the same thing, solving the same problem

1. Shared Services and Security Governance

Team Foundation has multiple lines of delineation that many times can be governed by size or number of users but that may not be the only reason to choose: TFS Instance, Team Project Collection, Team Project or Area to separate different teams. Design of standards used to evaluate teams and assign a container and security model will be required by 3M. The TFS application tier and the SQL Server hosting the TFS collection and data warehouse are dependent on the hardware for performance of these shared services.

* 1. Team Boundaries and Control

For the Enterprise Team Foundation Server (ETFS) 3M has defined which teams own the setup and support in the environment and who has control over other parts of the ETFS environment. The teams involved here are the DTA (*proposed*), release management (*proposed*) and project teams.

The Development Tools Administration (DTA) team sets up and supports everything in the blue boxes as shown in Figure 1 - Team Boundaries and Control.

The release management has control over executing builds for the UAT and Production build servers. The Release management team also controls the modification to the build definition for all project teams. The Release management team is responsible for setup and management of its tools interfacing with the builds servers and the ETFS.

The project teams control the Gated and CI build server(s) for executing CI builds as well as executing daily builds on the build server. The project teams can make modifications to the build definition on the CI build servers but must make requests for modifications to the daily build server, UAT build server and Production build servers. The project teams are responsible for the setup and management of tools interfacing with the build servers and the ETFS.

All tools must meet the minimum requirements of the version. In addition, a tool must have approval from the change board before interfacing with either the build servers or the ETFS environment.



Figure 1 - Team Boundaries and Control

* 1. Shared Services

The shared services for all TFS environments is the SQL Server, the team collections and team projects. Performance of those shared services will depend on the hardware but it also depends on the complexity and size of the project teams and the number of team collections and team projects. Below are some guidelines to help maximize performance of the shared services.

* + 1. TFS SQL Server

Performance and number of Team Project Collections and Team Projects is dependent on the configuration of the SQL Server. For example SQL Server performance depends on the amount of RAM the server has 8 GB to 64 GB. The performance of the TPCs and TPs that can be actively used concurrently is dependent on the amount of RAM available on SQL Server. A single SQL Server instance can have between 30 to 100 active TPC depending on SQL Server RAM. SQL Server disk space is another limiting factor as to how many TPCs and TPs can be stored on the server.

Microsoft’s recommendation is that the TFS SQL Server instance should be the only SQL server instance on that server. This is for performance reasons when hosting an Enterprise TFS system. The SQL Server Reporting Services (SSRS) and SQL Server Analysis Services (SSAS) should be on another server. The SharePoint server and Project server content SQL server databases should be on different servers also when integrated with TFS. Currently the ETFS at 3M is performing well. Recommendations to continue this state are included in this document.

* + 1. Team Project Collections

#### Single Team Project Collection

The recommendation for any organization is to start out with a single Team Project Collection (TPC). This provides for the simplest support and maintenance scenario. All Team Projects can share resources and within the collection can share artifacts such as work items and source code. Backups are for a single SQL database for the TPC. Other things to consider when choosing a single TPC is the number of team projects and the process templates that are used. A single TPC can, on average host, 250 CMMI team projects or 500 Agile/Scrum team projects before the SQL performance starts to degrade.

#### Multiple Team Project Collections

The number of TPCs depends on the business needs. If 3M needs to have teams segregated from other teams work than the recommend solution is to put that team or set of teams in its own TPC. Another reason for multiple TPCs is performance where 3M knows or plans to add hundreds of new Team Projects every year.

As a rule the more TPCs a TFS environment has the more support effort will be needed to do database backups, adding and managing users and modifying and updating process templates for each TPC. A side effect of multiple TPCs is that the sharing of source code and work items between teams is broken since every TPC has to be in its own database. An external solution such as NuGet is recommended in this case.

The TFS data warehouse does contain data from all TPCs and reports generated from the data warehouse can be affected if a TPC changes the meaning of measured data values.

* + 1. Team Projects

Microsoft’s recommendation is to have Team Projects align to a major application or solution, which may in turn align to a project team. But a project team could have multiple major applications so not all of the project team’s applications need to be in one Team Project.

The structure guideline for a TP is to have it laid out such that a developer can start modifying and building the application locally once it is retrieved from TFS source control. 3M should define and publish a standard structure guideline for the different types of technologies i.e. .NET, Java or Perl. This will allow developers to switch to different teams without having to reconfigure for the different teams.

* 1. Security

TFS, SharePoint, and SQL Server Reporting Services (SSRS) maintain independent data stores for groups, users, and permissions. Therefore security needs to be managed across all these systems with as little overhead as possible. The following sections provide some guidelines on how to best achieve that goal.

* + 1. Active Directory Users and Groups

Any change to an individual user of TFS should begin with the Active Directory (AD) users and computers Microsoft Management Console (MMC). Adding a user, changing a user’s password, disabling a user’s account—all of the same operations that would normally be conducted for user management—would still be done this way. Additionally, however, adding and deleting accounts must be done to each of the three applications (TFS, SSRS, and SharePoint). Again, the effort here can be reduced through the use of the [Team Foundation Server Administration Tool](http://www.codeplex.com/Wiki/View.aspx?ProjectName=TFSAdmin), but all three security data stores must be updated. This is a key reason managing users in TFS should be done via AD groups vs. an individual level.

The **Microsoft recommendation** is to have AD security groups defined for each of the TFS Project Collection groups and TFS Team Project groups.

* + 1. TFS Permissions

Active Directory groups are tied to individual permissions in TFS through TFS security groups. Server-level permissions are configured as TFS collection groups. The groups can be seen in the TFS project collection group membership screen. Team project level permissions are configured as TFS team project groups. The team project groups can be seen in the Team Project Settings group membership screen.

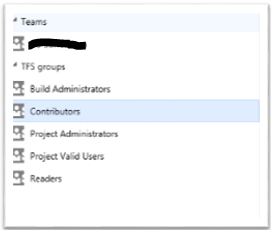


Figure 2 - TFS Team Project Groups

Members are added to these groups by nesting Active Directory groups or users inside of TFS groups. Where possible, this mapping should be done between TFS groups and Active Directory groups. Users should only be used as exceptions. TFS permissions can be set at the server level, the project level, the folder level, and the file level. Like other security implementations, it is easier to manage permissions if they are set higher in the hierarchy, using inheritance. Accessing the properties, at any of these levels, allows security to be managed through the security tab. In addition, each project comes, by default, with groups for readers, project administrators, and contributors. (NOTE: the groups and permissions may be customized via the “process templates” as needed.)

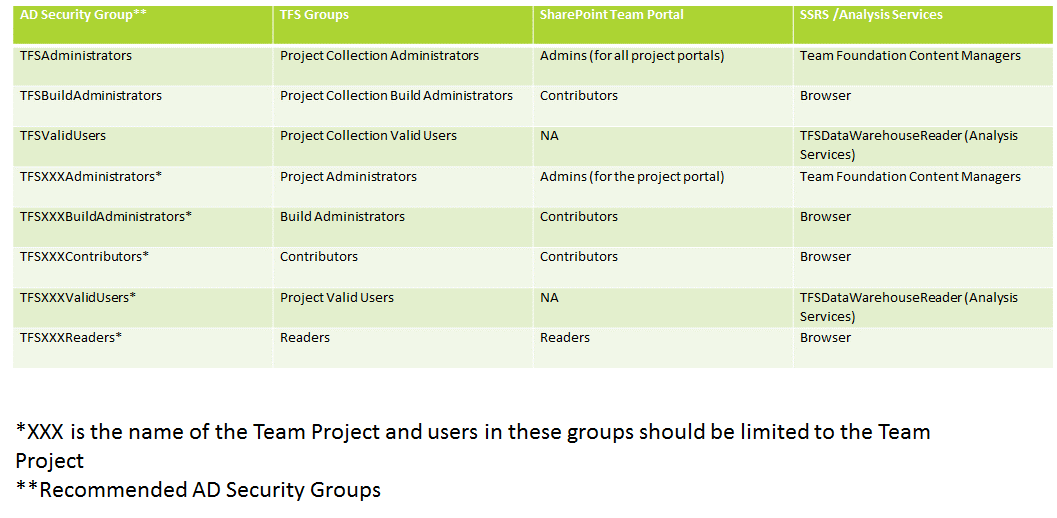


Figure 3 - Security Model Mapping

As noted earlier the default TFS team project groups cover broad areas of access. 3M has indicated that it needs tighter control of groups and access than what is provided in an out-of-the-box solution as shown in Figure 3 - Security Model Mapping.

The Microsoft recommendation for 3M is to limit the number of new team project groups and use the out-of-the-box groups as much as possible. 3M will have to apply these new team project groups into each process template it plans to use in TFS. (NOTE: Changes to a process template might not transfer on TFS updates or upgrades and may require manually applying these changes again.)

* + - 1. TFS Permissions for Scrum projects

Some questions have come up about the standard naming for Scrum contributors and how best to map these to TFS security groups. A simple way to handle this issue is to setup the appropriate groups as needed. For example, using the node “Team Members”, one can add subgroups such as Product Owners, etc. Then simply use the Tools/Process editor to add rules to work items to handle the state transitions.

* + 1. Source Control Security

Source control security is inherited by the folder structure in the team project as shown in Figure 4 - Source Control Security. From Figure 4 the ‘$’ is the server/collection level, “3M” is the team project level, “Main” is the branch or a folder level, and within a branch or folder is the file level. TFS permissions can be set at the server level, the project level, the folder level, or the file level. Like other security implementations, it is easier to manage permissions if they are set higher in the hierarchy, using inheritance. So where to break the inheritance to set the desired security control requires planning the structure of the branches and folders to minimize the number of manual changes to the inherited security control. Microsoft’s recommendation is to break the inheritance higher in the hierarchy for the desired security access for all children below that point.

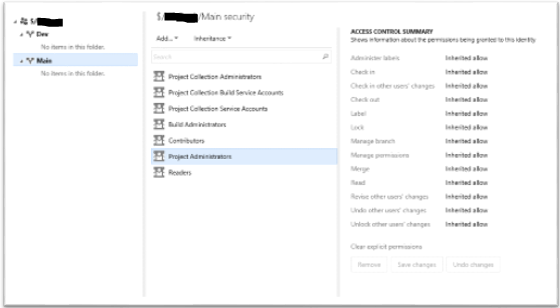


Figure 4 - Source Control Security

* + 1. Microsoft Office SharePoint® Server (SharePoint)

SharePoint has three security groups; Admins, Contributors and Readers. The recommended handling of these security groups is to map the AD TFS team project security groups to the level of access desired for SharePoint. Project Administrators or Dev Leads would be assigned Admin group for the project portal. The following would be added to the Contributors group; Developers, Project Managers, Business Analysis. Those that would be in the SharePoint Readers security group would be Readers in TFS also. Figure 3 - Security Model Mapping provides an example of mapping AD security groups to the different SharePoint security groups.

* + 1. SQL Server Reporting Services

SQL Server Reporting Services (SSRS) has two security groups; Team Foundation Content Managers and Browsers. Typically the Team Foundation Content Managers has the TFSAdministrators and the TFSXXXAdministrators as shown in Figure 3 - Security Model Mapping.

* + 1. SQL Server Analysis Services

SQL Server Analysis Services (SSAS) has one security group that all TFS users need to be part of; TFSDataWarehouseReader. If a person is not included in this group they will not be able to get results for Reports from SSRS. The recommendation is that an AD security group named TFSXXXValidUsers nests all the team project AD security groups and be added in to the TFSDataWareHouseReader group for SSAS on team project creation.

1. Process Governance

Process governance includes how the DTA team deals with project teams, management of the ETFS environment and meeting business requirements. Some areas include onboarding of project teams, handling of process customizations, tools for integration to TFS, and handling support issues.

Because every tenant in a multitenant TFS environment will desire unique processes they are likely to leverage Team Foundation Server’s ability to enact custom process templates. However since not only do all Team Project work item fields and elements flow to a single warehouse database and OLAP cube but all Team Project Collections converge in the same warehouse database and cube as well. In order to ensure that all teams cohabitate in this shared database *governance* needs to be applied to all customization of process templates.

* 1. Onboarding and Migrating

Every team that is going to be added or moved to TFS needs to be planned. 3M has to determine which teams are good candidates for TFS. Just like any other development project the 3M DTA team needs to use TFS and the Kanban board to plan what teams should be on-boarded and when. Planning when to migrate a project team into ETFS may depend on the project team’s release cycle.



Figure 5 - Onboarding and Migration Process Flow

The process for onboarding and migrating project teams into the ETFS needs to follow the process flow shown in Figure 5 - Onboarding and Migration Process Flow. The DTA manages the onboarding request throughout the onboarding process flow. The process starts with a request from a project team and a Backlog Item is created in the DTA team project. Someone is assigned to interview the project team to determine size and complexity, development processes, branching and merging models and etc… Based on this information DTA and the change board review the request and assign it a priority if approved. Once approved the migration process will start where DTA does a test migration into the TFS test environment. Once the test environment is approved by the project team and documented by DTA the final migration to production can take place.

|  |  |  |
| --- | --- | --- |
| Effort | Project Team Type | Comments |
| High | Large project teams or projects with large and/or complex components | These are project teams that can take many iterations to migrated in to TFS |
| Medium | Simple project with non-complex components or small project teams | These project teams should take 1 or 2 iterations to migrated in to TFS |
| Low | New project teams with no projects to be migrated in to TFS | These project teams should take less than 1 iteration to add to TFS |

The size and complexity of a project team will determine how long to implement and the effort for onboarding them to TFS. Teams that have small and straightforward project structures can be done within a four week iteration as an example. Teams that need to rework the project structure, meet minimum migration requirements, have a large footprint or large number of components will most likely take multiple iterations.

Project teams that are requesting a new application team project are the easiest to onboard since no planning of moving of existing code and files is required. The project team just needs to pick a process template and a branching and merging option.

To help with the onboarding of a team the DTA team should have a test TFS environment that is similar to the Enterprise TFS environment. Before adding the project team to the production ETFS a test onboarding should be done to determine if the branching and folder structure allows the team to manage and build their code. This also helps to determine which files need to be ported, and which ones can be archived.

To simplify the process of onboarding a team the latest version of the source control files should be the only thing moved into the new ETFS environment. The old source code is then archived to save the past history if needed for audits.

During the onboarding and migration process a project team may go dark, i.e. stop communicating or completing documents as needed. In order to ensure DTA resources are not overwhelmed it will be necessary to put these projects teams back into the queue and re-prioritized and basically restart the onboarding and migration processes for that project team. As to how long 3M allows a project team to be dark and re-queued is a matter for DTA and the change board to decide.

Each team needs to pick the process template it plans to use and a branching model. As for the choices a team has it should be limited to three process templates and three branching options that 3M can support. When it comes to a team choosing a branching option other than the default one the project team needs to justify why they want that option. **Warning**: If a project team decides later that it does not want to use the process it picked during the on-boarding process it will have to redo the on-boarding process and this means losing all of its work item history.

* 1. Branching and Merging Models

3M needs to limit the number of options for branch and merging models that a team project can use. Otherwise support for issues will exceed the 3M IT team resource capacity. By 3M standardizing on a branching and merging model it will allow developers to switch between teams without having to learn new processes related to branching and merging. More details about recommended branching and merging options can be found in the ALM Rangers guide titled “[Version Control Guide](http://vsarbranchingguide.codeplex.com/)”.

3M is considering use of the three branching and merging models as outlined is the following sections.

* + 1. Default Branching and Merging Option

This default branching and merging option is the Microsoft recommended model to have at the start of all team projects. This provides for a straight forward set of branches that can be managed by any project team with easy merging as needed from changes to releases of an application. From this option another branching and merging options can be easily added to this baseline as a project team matures in its Application Lifecycle Management (ALM) processes.

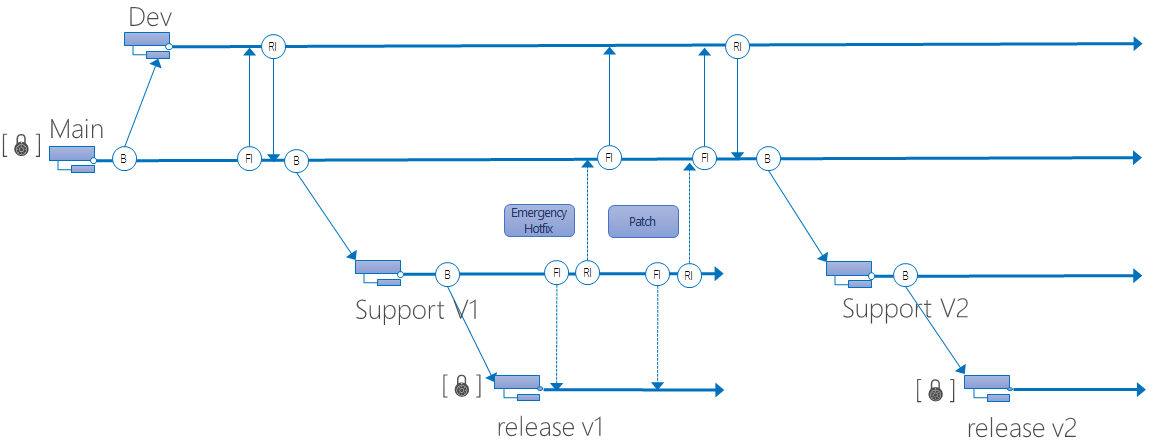


Figure 6 – Default Branching and Merging Option

* + 1. Dual Project Development Branching and Merging Option

This branching and merging option expands the default one by just allowing a second concurrent development project branch. The complexity of this branching and merging model is keeping the two project teams in sync with changes promoted to the Main branch. Microsoft’s recommendation is that a team has to justify the need for the more complex option and show it has a mature understanding of managing this branching and merging model. This branching and merging model is one that can be done temporarily and then go back to the default option as needed.

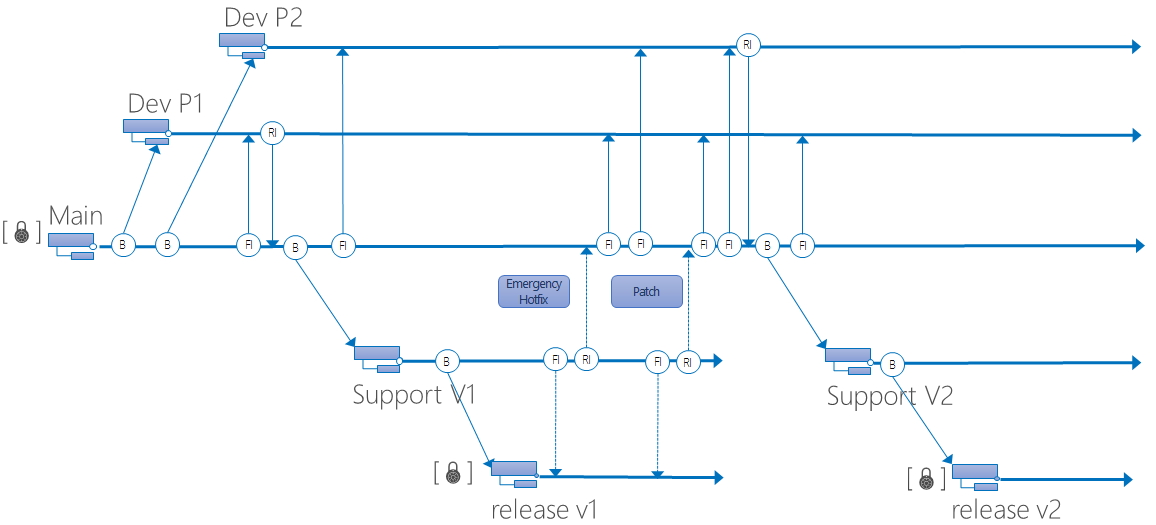


Figure 7 - Dual Project Development

* + 1. Parallel Project Teams Branching and Merging Option

This branching and merging option allows for parallel project teams. The risks for a project team are that the two team branches can become so out-of-sync, so merge conflicts make it almost impossible to manage and have a clean set of branches including Main that will build. It is important that all the branches are sync’d often to ensure the number of merge conflicts are kept to a minimum. Another trap to watch out for is if a project team does baseless merges to fix the merge conflicts. Doing baseless merges is a sign that the branching model is not working for the project team.

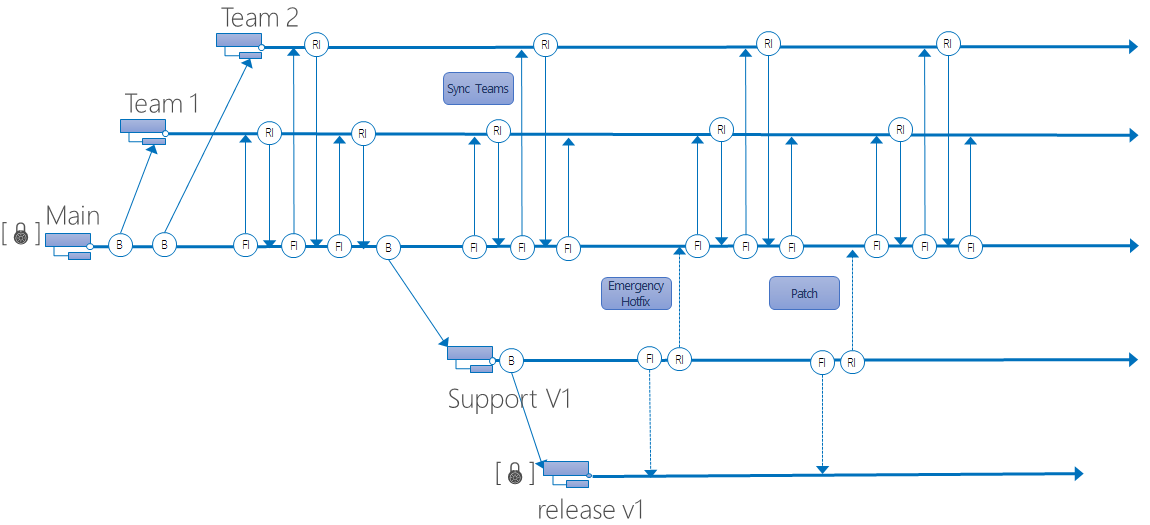


Figure 8 - Parallel Project Teams

This branching and merging option can work for a disciplined project team that performs frequent sync merges and controls which teams is changing areas of code in the application. The Microsoft recommendation is a project team needs to show why they need to use this branching and merging option and show a plan on how they will keep all branches in sync to prevent baseless merges and minimize merge conflicts.

* 1. Tool Integration

Microsoft’s recommendation for allowing third party tools or home grown tools to integrate with TFS, is to put the tools through a review and a final approval from the change board. 3M needs to understand how the tools will change possible critical data in the TFS databases and data warehouse before using them. 3M may also need to understand if the integration has to be used during the build and what affect it has on that build process. Also processes need to be defined as to when and how to handle upgrades from integration tools.

* 1. TFS Updates and Upgrades

As with any application updates or upgrades they need to be planned and go through a review and get approval from the change board. In reviewing the updates, changes to process templates will need to be carefully reviewed to determine if those changes have to be reapplied to the process template after the update. For 3M to be successful in an update or upgrade they should use the test Enterprise TFS environment to test and practice before applying to the production system. Other areas to watch for in updates is where a project team has changed or added required fields to its report and process templates.

* 1. TFS Process Templates

It is recommended that 3M use the current out-of-the-box process templates and only make changes related to the business required TFS security roles for team projects. 3M should limit the number of process templates it supports due to the limited IT resources for managing the Enterprise TFS.

If 3M needs to modify template process fields and work flows they should be limited changes and be done before bringing too many teams into the ETFS. Every time a process template is made those changes need to be applied to current TFS TPs which is a manual process.

Any process template changes should be reviewed and approved by the change board. The reason for this is that all current and future teams will be affected by these changes. The changes can also affect the meaning of the data in the TFS data warehouse. In addition, any updates or upgrades to newer versions of TFS may require that the changes be reapplied to the new process templates. If that is the case having the changes formally documented and approved will ensure that all agreed upon changes become part of the upgraded TFS environment.

When considering customizations for a process template or work item types the following Table 1 summarizes the possible customizations and possible risks and effort.

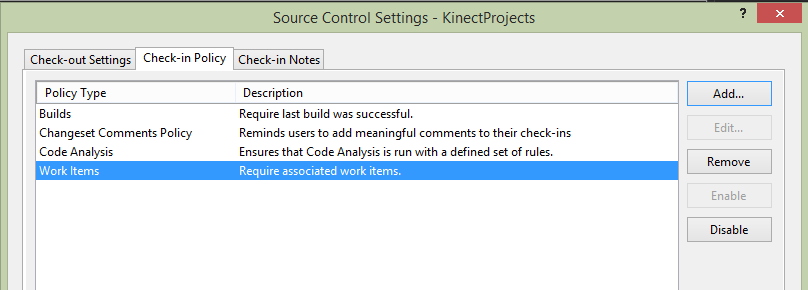
|  |  |  |  |
| --- | --- | --- | --- |
| Customization | Effort | Comments | Risks |
| Adding new field to a work item | Medium | If reporting is required on this new field will require new or changed reports | When TFS is updated or upgraded might have to updated the new or change reports and may have to reapply the new field to process templates or work item types |
| Changing the meaning of a work item field | Medium | Easy to do but will impact reporting results | Reports generated from the data warehouse may not be as usefully or report unusual results |
| Add new state transitions to a work item | High | Adding new state transitions can be hard to debug and prevent a process template or work item type from be uploaded to TFS | On TFS updates or upgrades will most likely require the changes be reapplied to the process template and work item type |
| Adding or changing required fields to a work item | High | Adding or changing a required field can be hard to debug and prevent a process template or work item type from be uploaded to TFS | On TFS updates or upgrades will most likely require the changes be reapplied to the process template and work item type |
| Adding a new work item type | Medium | Hard to debug if the TFS does not allow the work item type to be uploaded to TFS | The new work item type will exist after a TFS update or upgrade, but may require modification if based on another work item type. |
| Adding security customizations | Low | Simple to do in the process template | Low risk |
| Adding or changing Work Item Queries | Low | Simple to do in the process template | Low risk |

Table 1 - Process Template Customization Risks

Standalone process template changes for a single team project should be avoided for the following reasons;

* Requires management of changes to multiple templates
* Updates or upgrades could break the modified template and the DTA team has to reapply these changes on every update or upgrade for standalone process templates.
* Could change the data warehouse data and cause issues in TFS reports and dashboards.
  1. Checkin Policies

TFS 2013 Update 4 supports four policies out of the box:

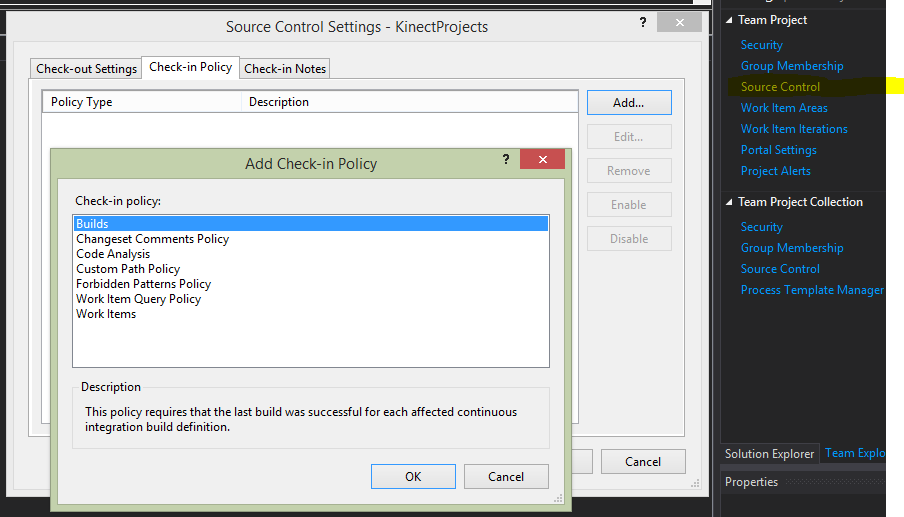


If is strongly recommended that 3M projects have “Changeset comments Policy and Work Items turned on as a minimum. Consider adding Code Analysis and Builds as teams become more sophisticated. StyleCop and other related tools can also be added as part of the build process.

When using the latest TFS power tools, the following three policies are added:

|  |  |
| --- | --- |
| **Policy** | **Description** |
| Custom Path Policy | This policy scopes other policies to specific folders or file types |
| Forbidden Patterns Policy | This policy prevents users from checking in files with fobidden filename patterns |
| Work Item Query Policy | This policy allows you to specify a work item query whose results will be the only legal work items for a check-in to be associated with |

The following picture shows all policies:



* 1. Change Board

A change board would be similar to an ITIL Change Advisory Board. The purpose is to review and triage requests for TFS process changes and TFS template changes. The other responsibilities of the change board is to review and prioritize the onboarding of project teams in to the Enterprise TFS, and to be included in the planning for updates or upgrades of TFS to ensure that changes do not conflict with critical TFS operations. The change board should also review and prioritize high effort support issues submitted to the DTA. In some cases the change board does need to reject change requests.

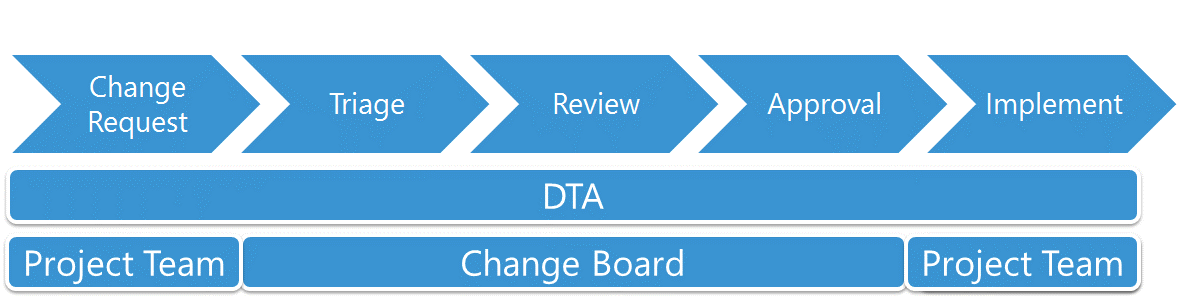


Figure 9 - TFS Change Request Process Flow

When project teams want to change their process templates or just suggest changes to the ETFS processes in general, they need to submit a change request which becomes a backlog item in the DTA team project. The DTA will triage the change request to determine complexity and effort, plus any risks to supporting the change in the ETFS. Then the change request is reviewed with the project team, DTA and the change board. If the change request is something that the change board agrees would benefit the ETFS and-or project teams, then the change request should be approved. Otherwise they should send it back for further investigation or with suggested changes.

The change board can also deny the change request. Some of the reasons for rejecting a change requests is that it does not help 3M or the project teams improve software quality, or reduce or keep support cost inline for the ETFS. Once approved the DTA team will implement the change request and get sign-off from the project team and change board.

The change board should have members from the following areas of 3M; management, security, IT infrastructure, architects, and developers. The change board should meet at as needed, but should meet every two months at a minimum.

All work for the TFS environment should use a TFS team project to manage the work items related to requests and approved changes. The Kanban board should be used during the change board meeting to visually see the state of all the work tasks and backlog items. The Kanban board will also help the change board manage the TFS resources to see where a resource is overloaded.

1. Build Service Governance

Build environments are very unique to each individual software project team and many times requires high touch from the project team itself. This project will design a governance approach that can scale to the enterprise. What this means is designing a shared build environment that can be used by small teams or teams that have generic needs as well as a governance model that allows larger teams and teams that have unique needs to manage their own build server farm while still using the Enterprise TFS service in a standard way.

* 1. Continuous Integration Build Servers

The Continuous Integration (CI) build server can and should be a shared resource for project team builds, unless a project team has special build components that require its own CI build server. If the build queue is getting too deep a set of CI build servers can be set up to handle the load and spread the CI builds out among them. 3M is also considering the use of virtual on demand build servers. 3M could monitor the build queue and spin up a CI build server and redirect builds in a round-robin fashion using RTC.

* 1. Daily Build Servers

3M is planning to use a daily build server to perform the UAT and production builds. As with the CI build servers these can be shared for the different project teams, unless special build components are required to build the application, than the project team needs its own daily build server.

Along with building the application 3M should consider executing Build Verification Tests (BVT) to help determine as early as possible if the build is broken. The use of BVTs will help prevent wasted effort of the testing team on broken builds.

3M is considering the use of virtual on demand build servers for its UAT and production builds. This will require that the build definition be programmatically modified to support the different build server targets.

* 1. Build Definition

When a project team is added or migrated to the ETFS, it should get a set of build templates to start with based on its application needs. Optionally, build templates could be provided to include CI builds with unit testing and code analysis, and also with Build Verification Tests (BVT). The team project build engineers can modify the CI build definition as needed for their use. Modification of build templates need to be submitted as a change request to be reviewed and approved by the change board. The project team is responsible for requesting the UAT or production builds as needed.

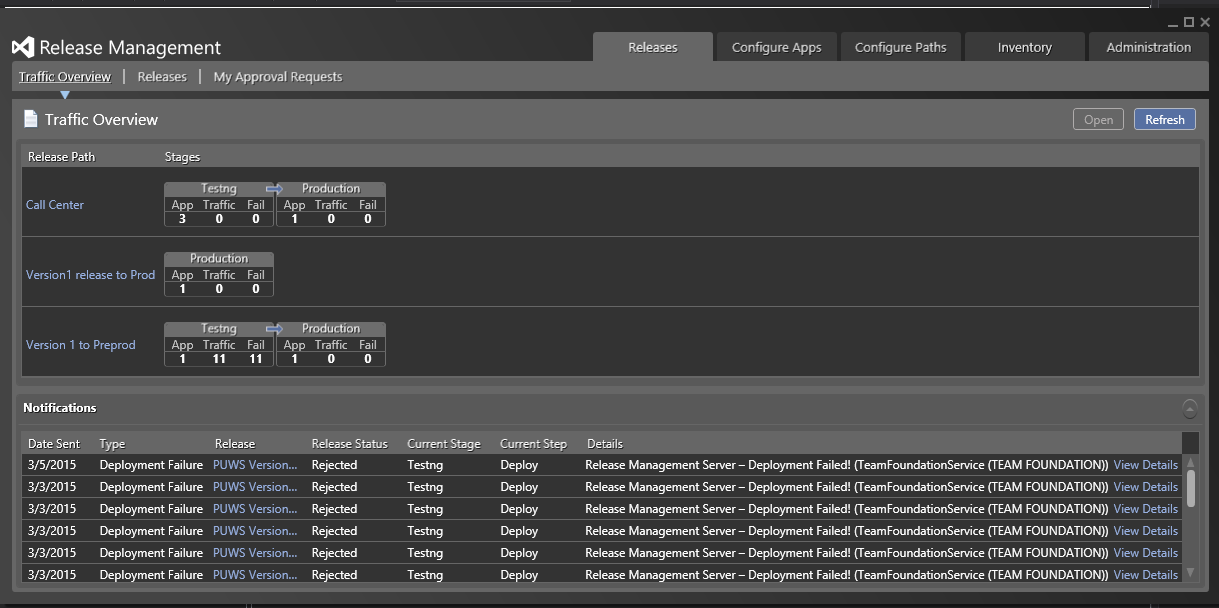
There are options that can be defined for gated builds and check-ins, but they need to be defined per TP as needed. If these options are enabled it should also be enabled for the CI build definitions. The same rules would apply for code coverage requirements using unit tests or static code analysis.

* 1. Symbol Server

While this section says “server”, in 3M’s case it will be a commonly accessible file-share where critical information necessary to debug code issues will be stored. TFS takes care of adding and removing the symbols as part of the build process. It is critical for debugging applications in all environments including production, that this capability be enabled as part of every build (particularly production). This [link](http://blogs.msdn.com/b/jimlamb/archive/2009/06/15/symbol-and-source-server-in-tfs-2010.aspx) gives a good introduction to the process.

* 1. Release Management

Release Management has always been a critical part of the development process. Until fairly recently, it has been somewhat left as an “exercise for the build master” in TFS. Now the latest releases of TFS 2013 has included [Release Management](https://www.visualstudio.com/en-us/explore/release-management-vs.aspx) as a full-fledged part of the build process. As can be seen below, the process has an entire tool/workflow devoted to it:



While this is primarily the responsibility of the build master to setup (and permission-wise) they are the only ones who can, the workflow when deploying involves the whole team.

1. Integration Governance

Because Team Foundation Server is the central hub to the entire software project teams and contains many spokes of integration it’s very easy to integrate with TFS. With that ease of integration comes the need to be thoughtful and deliberate in taking on dependencies to TFS to enable changes and upgrades to remain possible. Use Microsoft guidelines and governance planning for these other 3rd party systems.

* 1. Third Party Tools or Home Grown Tools

Any tool that wishes to integrate with the ETFS needs to go through a review and then be approved by the change board. In reviewing a tool for integration, an important thing to look for is the modification of critical TFS data in the database and data warehouse. Is 3M willing to accept the risk of that data being modified and is the tool needed for improving the business processes of managing TFS and the team projects. If the tool is just reading data from the TFS database or data warehouse the risk is low and may not need to go through the review and approval process.

Tools that use service accounts to gain access have to be monitored to ensure that those accounts are not added to TFS AD security groups or roles that would give them any kind of access for modification.

When an integration tool that interfaces with the ETFS has an update or upgrade the project team needs to submit a change request to the change board with the details of the upgrade for possible database changes and process changes. The tool needs to have a rollback plan if the deployment fails or causes outages. The DTA team needs to ensure a backup of the TFS data is captured for worse case issues and can recover TFS. Once the update or upgrade is approved, the work can be scheduled in the Kanban board.

When updates or upgrades are planned for the ETFS all integration tools need to be informed and offered the chance to indicate if this change will break the tool integration. If a breaking change can occur the tool should provide a plan of how it can be updated to work with the TFS update so delays for updates of the ETFS is kept to a minimum.

SharePoint and Project Server are supported integration tools to TFS and can provide documentation, reporting and project management to 3M. But like any of the other tools they need to follow the processes, reviews and approvals of the change board before being integrated.

1. Support Governance
   1. Team Foundation Server Support

Support for Team Foundation Server (TFS) should be limited to just the areas that involve what was installed for TFS and the Team Project Collections (TPC) and Team Projects (TP). For TFS this is the application tier, build servers, data tier (SQL Server TFS databases and data warehouse), SSRS and SSAS for reporting. The TPC and TP involves onboarding and migration of project teams and process template setup and modification. The support effort is broken down into the following; planned work, maintenance and unplanned work. All of this work should be tracked in a DTA team project as work items and backlog items. This way review of current work and queued work can be done via a Kanban board for the DTA team and the change board.

The DTA is responsible for the support and maintenance of the TFS environment which includes the application tier and the build servers. At 3M the SQL team is responsible for the data tier and SSRS and SSAS.

* 1. Customization Policies

All customization from either the project teams or the business need to be submitted to the DTA team as a change request. The change request is managed in the DTA’s TP as a backlog item. The DTA team will review the change request to determine if the customization will benefit all ETFS team projects or just the requestor. The review will identify the effort needed to implement the change request and the impact to support and maintain the customization. These findings will be presented to the change board and it will assign a priority to the backlog item if approved.

In TFS the areas allowed for customization are the following; security, ETFS policies, process templates, build templates, branching and merging model, integrations and reports. 3M and the change board may decide that some of these areas will not be allowed to change after a TPC or TP is set up for a project team.

* 1. Service Level Agreements

A Service Level Agreement (SLA) for this document is not meant to define a hard set of time for work, but instead will provide guidance of what should be covered with an SLA. Some of the services have been covered in detail in other sections of this document and should be reviewed. The following list are services that the DTA team should provide and define a SLA;

* On boarding a new project team project into TFS
* Migration of an existing project team’s project into TFS
* Approval process for process template changes including steps to get approval and time frame for decision.
* Time frame for implementing process template changes if approved
* Approval process for integration tools including steps to get approval and time frame for approval
* Policies and time frames for unplanned work items or support tickets
* Maintenance tasks with server down time
* TPC and TP storage limits
* Performance of build servers
* Build Server diagnosis
* Processes and time frame for requesting build definitions for build server
* Recovery, load balancing and failover strategies of ETFS
* Rules for inactive TPC or TP
* User Access request process and time frame and also validate roles for ETFS
* Defined supported development tools and versions
* Operation-level agreement, which specifies which teams perform which operations and how frequently i.e. SQL Server backups
  1. Team Foundation Server Monitoring

To help maintain and provide the best experience of the ETFS environment, it is recommended that 3M monitor the TFS environment and servers. Microsoft provides a solution using System Center Operations Manager (SCOM) and Team Foundation Server 2013 Management Pack (MP).

The TFS MP monitors the TFS application tier, the build servers, data warehouse, TPCs and TPs. It provides a management console that lists all the TFS related objects that it discovered during the setup of the MP. Since some of the objects are dependent on other TFS objects the MP in the management console will provide a health roll up for health status. Alerts can be defined to inform the support team of issues or events that are being monitored by the MP. 3M has installed this and integrated it with SCOM. It is currently being tuned for 3M’s unique environment (early 2015).

The following objects are monitored by the MP;

* Event logs
* Queries to the web services for responses
* TFS performance counter.

Following is a list of TFS Service performance counters that are monitored for SQL server performance and should be monitored by 3M;

* Total Number of Failed retry sequences
* Total Number of Throttling Events
* Total Number of SQL Batches
* Current SQL Executions/Sec
* Current SQL Notification Queries/Sec
* Current Task Executed/Sec
* Active Team Project Collection Service Hosts
* Active Application Service Hosts
* Active Deployment Service Hosts
* Current SQL Execution Retries/Sec
* Current SQL Connection Retries/Sec
* Current SQL Connection Failures/Sec
* Average SQL Connect Time
* Active SQL Connections
* Average Response Time
* Current Requests/Sec
* Current Server Requests

Greater details about the TFS performance counter which were introduced starting with TFS 2012 can be found in the following article [New Performance Counters with TFS](http://blog.jimszubryt.com/2012/05/21/new-perfmon-counters-with-tfs-11-beta/).

* 1. Governance and ETFS Training

In order for any governance to be successful and adopted by its user community a company needs to provide information and training about that governance, the processes and the tools. 3M should consider the following for training and publishing information.

|  |  |
| --- | --- |
| Provide | Why? |
| Training for the products and services | Training and education about Enterprise Team Foundation Server in the governance plan helps drive adoption and helps reduce support costs. |
| Education about the governance policies | Training the user community appropriately increases compliance with the policies, increases satisfaction with the services, and reduces support costs. |
| Content to support services and policies | Having good quality resources and information available helps users find the answers when they have questions about a service, process, or policy. |
| A good search infrastructure | Having a good search infrastructure helps users find what they need when they need it. |

1. Executive Summary

3M has many team projects from different divisions. Each of this divisions have slightly different Application Lifecycle Management processes. The issues faced from these divergent environments is the inefficient cost of supporting and maintaining these standalone systems.

The risks to 3M are missed audits of unknown applications, reinvented components or purchase of duplicate tools and developer resources having to learn new ALM processes when switching teams, and high support costs of a source control environment.

Governance recommendations;

* Use Active Directory for security management
* Limit the number of Team Project Collections
* Have consistent Application Lifecycle Management process
* Control process changes to benefit all project teams in the Enterprise Team Foundation Server environment
* Manage and plan project teams on-boarding and migration to Enterprise Team Foundation Server
* Manage the use of build servers
* Provide processes and change controls for allowing tools to integrate with the Enterprise Team Foundation Server environment
* Define Service Level Agreements

The recommended governance in this document provides the following benefits;

* Better security management via Active Directory and common Team Foundation Server security roles
* Common set of ALM processes (CMMI, Agile, Scrum) to benefit project teams in the Enterprise Team Foundation Server
* Change board and a process for change management to support and improve the project teams
* Consistent audits across applications managed within the Enterprise Team Foundation Server
* Development resources able to help or move to other development efforts without retraining
* Visibility into component reuse and sharing of development tools
* Reduced support and maintenance cost of source control and Application Lifecycle Management tools for project teams in the Enterprise Team Foundation Server environment

Using these governance guidelines will provide 3M the tools to manage the Enterprise Team Foundation Server for planned work, maintenance work and unplanned work, while controlling the effort and cost currently not possible with today’s standalone source control environments.

1. Next Steps

All divisions will continue to use ETFS and/or migrate from any other existing ALM system. Migrate a select set of project teams of various processes and complexity for the first six months using the proposed system. This will help the DTA team learn and define a process for on-boarding and migrating project teams. During these six months the change board will learn what process changes should be considered for all future project teams in the ETFS, while learning how to define change request processes and setting priorities for bring project teams into ETFS.

Leverage Microsoft to help setup the new Enterprise Team Foundation Server processes and rules for on-boarding and migration of project teams. Also use Microsoft to help define processes for change requests and testing of the change requests. Microsoft can coach the change board on how to prioritize work for the DTA.

Look at having Microsoft Premiere Developer perform a TFS health check after a major release or, at a minimum on a quarterly bases. A baseline health check should be performed after every ETFS enhancement.

Look at having Microsoft perform an ALM maturity assessment in 12 to 18 months just as this effort was performed in late 2014. This will provide a good summary of what is working well and what could be improved. The assessment will provide a roadmap of how to improve the ALM processes, usage and support of TFS.

1. Resources and References
   1. Shared Services and Security

* Team Foundation Server Security Architecture

(<http://msdn2.microsoft.com/en-us/library/ms252473.aspx)>

* Team Foundation Server Default Groups, Permissions, and Roles  
  (<http://msdn2.microsoft.com/en-us/library/ms253077.aspx>)
* Team Project Collections and Management

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

* 1. Processes
* TFS Planning Guide

<http://vsarplanningguide.codeplex.com/>

* Version Control Guide (ex-Branching and Merging)

<http://vsarbranchingguide.codeplex.com/>

* TFS Release Management (2012)

<http://msdn.microsoft.com/en-us/library/dn449957.aspx>

* Customize a process template

<http://msdn.microsoft.com/en-us/library/ms243782.aspx>

* New Features in Team Build 2013 – TFS 2013

<http://www.dotnetcurry.com/showarticle.aspx?ID=964>

* Customize functional areas within a process template

<http://msdn.microsoft.com/en-us/library/ms400734.aspx>

* Overview of process template files

<http://msdn.microsoft.com/en-us/library/ms243856.aspx>

* Work with team project artifacts, choose a process template

<http://msdn.microsoft.com/en-us/library/ms400752.aspx>

* Define the initial configuration of Team Foundation version control

<http://msdn.microsoft.com/en-us/library/ms243831.aspx>

* 1. Build Services
* Scale out the build system

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

* Installing and Configuring Test Agents and Test Controllers (2012)

<http://msdn.microsoft.com/en-us/library/dd648127(v=vs.110).aspx>

* Select a staging location and set up a drop folder

<http://msdn.microsoft.com/en-us/library/bb778394.aspx>

* TFS 2013 Build Extensions

<http://visualstudiogallery.msdn.microsoft.com/2011f516-15a7-4f9a-8b86-1e0894a75739>

* Setting up an Ant or Maven build in TFS

<http://msdn.microsoft.com/en-us/library/jj155784.aspx>

* Customize the build process template

<http://msdn.microsoft.com/en-us/library/dd647551.aspx>

* New Features in Team Build 2013 – TFS 2013

<http://www.dotnetcurry.com/showarticle.aspx?ID=964>

* Create and Work with a Custom Build Process Template

<http://msdn.microsoft.com/en-us/library/dd647551(v=vs.110).aspx#regsiter_template>

* Customize the build process template

<http://msdn.microsoft.com/en-us/library/dd647551(v=vs.120).aspx>

* Create or edit a build definition

<http://msdn.microsoft.com/en-us/library/ms181716(v=vs.120).aspx>

* How to modify the TFS 2013 default build process template (GitTemplate.12.xaml & TfvcTemplate.12.xaml) to enable releasing from a build

<http://blogs.msdn.com/b/visualstudioalm/archive/2013/12/09/how-to-modify-the-build-process-template-to-use-the-option-trigger-release-from-build.aspx>

* 1. Integration
* TFS 2013 Object Model Installer

<http://visualstudiogallery.msdn.microsoft.com/3278bfa7-64a7-4a75-b0da-ec4ccb8d21b6>

* Extending Team Foundation

<http://msdn.microsoft.com/en-us/library/bb130146.aspx>

* Using the TFS Client Object Model

<http://msdn.microsoft.com/en-us/magazine/jj553516.aspx>

* Version Control in the TFS Client Object Model

<http://msdn.microsoft.com/en-us/magazine/jj883959.aspx>

* PowerShell TFS 2013 API

<http://nakedalm.com/powershell-tfs-2013-api-1-get-tfscollection-and-tfs-services/>

* Microsoft Visual Studio TFS 2013 Power Tools

<http://visualstudiogallery.msdn.microsoft.com/f017b10c-02b4-4d6d-9845-58a06545627f>

* Extending Work Item Tracking by Using the Client Object Model for Team Foundation

<http://msdn.microsoft.com/en-us/library/bb130347.aspx>

* 1. Support
* New Performance Counters With TFS 11

<http://blog.jimszubryt.com/2012/05/21/new-perfmon-counters-with-tfs-11-beta/>

* SCOM Management Pack for Team Foundation Server 2013 Download

<http://www.microsoft.com/en-us/download/details.aspx?id=41696>

* SCOM Management Pack for TFS 2013 Available

<http://blogs.msdn.com/b/bharry/archive/2014/02/05/scom-management-pack-for-tfs-2013-available.aspx>

* SCOM - New Management Pack for Team Foundation Server 2013 Available

<http://kevingreeneitblog.blogspot.com/2014/02/scom-download-new-management-pack-for.html>

## ALM Assessments

Please see each teams spreadsheet for more details. The summary scores for both HIS and IPD are included following each sections description. The detailed questions in the assessment were used to produce these scores, the following sections listed herein are designed to show the typical results an organization will reap from raising their score in a particular area.

### Architecture & Design

#### Basic

* Architecture not properly documented
* Inconsistent or non-existent use of Modelling tools
* No clear process to transform business requirements into technical requirements
* Architecting not considering Deployment early in design
* Unclear understanding of the Architect Role

#### Standardized

* Architecture role understood and clearly identified, combined with other roles
* Tools identified, early adoption phase
* Some habits starting forming, some process consistency
* Documentation irregularly maintained, not consistently across teams or projects

#### Advanced

* Dedicated architecture team
* Architectural tools take into account the deployment process
* Integrated tools used across different teams across different projects
* Leverage use of practices and processes
* Application of practices and processes across teams and projects

#### Dynamic

* Formalized, documented architected process
* Consistent inclusion of patterns and practices
* Clearly defined mechanism to share or force usage of patterns and practices across projects and teams.
* Contributes back to the development community internally and externally through the use of published articles, whitepapers and conferences

#### HIS Scores:

With a summary total of **30**, HIS can be considered **Advanced**.

#### IPD Scores:

With a summary total of **20**, IPD can be considered **Standardized**.

### Software Configuration Management

#### Basic

* May or may not be using Source Control
* Local copies of code
* Manual on-demand build process
* Build process not documented
* No traceability between build and content/work performed and requirement
* Unclear understanding of branch/merge concepts
* Irregular check-ins

#### Standardized

* Non-integrated source control tool usage
* Dedicated build machine
* Informal undocumented build process
* Branching and merging understood by lead integrators
* Daily or regular check-ins performed

#### Advanced

* Use of an IDE integrated source control tool
* Dedicated configuration management role
* Formal, documented build process
* Build metrics published regularly
* Build on demand enabled
* Unit tests run after BVT

#### Dynamic

* Highly sophisticated build scripts
* Integration of multiple internally and externally produced code modules
* Build outcome Alerting and monitoring

#### HIS Scores:

With a summary total of **30**, HIS can be considered **Advanced**.

#### IPD Scores:

With a summary total of **10**, IPD can be considered **Basic**.

### Governance

#### Basic

* Projects started with limited justification
* Projects funded on key influencer opinions
* No ROI evaluation or retrospective.
* No Portfolio review process
* No compliance program or target in place
* No process improvement initiative in place

#### Standardized

* Informal certification for chosen compliance program
* Compliance certification applied and monitored inconsistently across teams.
* Semi-manual tools processes (Excel lists etc.)
* Random use of initiative targets

#### Advanced

* Formal certification for chosen compliance program
* Using portfolio management techniques but portfolio and project management tools not necessarily integrated
* Cross–team resources managed and time assigned
* Integrated with certification and compliance program

#### Dynamic

* Fully integrated portfolio management tools & process
* MS Portfolio Manager integrated with project management system and development system
* Participation in creation and review process of Industry standard compliance programs
* ROI and retrospective supported by metrics

#### HIS Scores:

With a summary total of **10**, HIS can be considered **Basic**.

#### IPD Scores:

With a summary total of **10**, IPD can be considered **Basic**.

### Deployment & Operations

#### Basic

* Little or no communication between operations and development teams
* No formal help desk/bug tracking process
* No tools, email based, manual follow-up
* Infrastructure deployment issues identified and resolved at deployment time
* No segregated environments, such as development, pre-production, test, UAT, production)
* Ad-hoc Build promotion schedule
* Unregulated build promotion
* Undocumented environment

#### Standardized

* Stand alone Help desk incident tracking tool (training, user issues, infrastructure)
* Non-integrated Bug tracking
* Some monitoring in place
* Some procedures and/or approval process in place for build deployment
* Limited automation and validation of build deployment
* Deployment Manager role identified
* Documented infrastructure
* Segregated environment but ownership unclear

#### Advanced

* Help desk integrated with bug management
* Monitored instrumentation hooked into infrastructure and applications.
* Tools to deploy and validate successful build deployment, smoke tests, testing scripts, data generation scripts
* Approval process with traceability integrated
* Clear cross-functional team identified
* Infrastructure Architecture documented in integrated tools
* Segregated environments, ownership clearly defined, promotion procedures well understood and consistent between environments.

#### Dynamic

* Using Helpdesk quality metrics on turn-around time, cost of maintenance and identification of error prone subsystems
* Automated Deployment
* Pro-active ongoing monitoring

#### HIS Scores:

With a summary total of **10**, HIS can be considered **Basic**.

#### IPD Scores:

With a summary total of **10**, IPD can be considered **Basic**.

### Testing & Quality Assurance

#### Basic

* No dedicated Q/A team
* Ad-hoc functional testing performed by development team. Closer to debugging than testing.
* No quality metrics
* Long fix and deploy cycles
* High regression bug rate

#### Standardized

* Dedicated Q/A group staffed
* Test Plan process defined
* Un-integrated testing tools in place
* Test procedures and environment informally documented
* Rudimentary progress tracking

#### Advanced

* Organization culture accepting of defined testing policies
* Test planning begins at the requirements phase
* Testing is measured and quantifiable process
* Integrated tools generating publishable metrics

#### Dynamic

* Test Process improvement group and tools in place
* Industry leadership on evaluating potential testing tools and strategies
* Defect prevention practiced
* Testing based on statistical sampling, measurement of confidence, trustworthiness and reliability

#### HIS Scores:

With a summary total of **20**, HIS can be considered **Standardized**.

#### IPD Scores:

With a summary total of **10**, IPD can be considered **Basic**.

### Project Planning & Management

#### Basic

* No formal stakeholder communication plan in place
* Informal or non-existent processes for estimation, planning, risk management, and scope. Gut-feel approach.
* Informal team coordination and task assignment using email, or verbal
* Financials not evaluated by P.M. on an ongoing basis.
* No clear P.M. defined responsibility

#### Standardized

* Individual un-integrated non-standardized use of Project planning tooling
* Tool usage dependent on strength of individual P.M.
* Financial manually evaluated manually by P.M.
* PM responsibility clearly assigned

#### Advanced

* Integrated management of bugs, tasks, change requests
* Use of EPM for financial and resource tracking through EPM to VSTS integration
* External resources, stakeholders and partners sharing project information and have integrated tools to perform their role in the project (Sharepoint, Team Plain)
* Dedicated Project Managers

#### Dynamic

* Portfolio Management, Project management have full integration
* Metrics used to drive project and aide in estimation and re-estimation
* PMO in place

#### HIS Scores:

With a summary total of **10**, HIS can be considered **Basic**.

#### IPD Scores:

With a summary total of **10**, IPD can be considered **Basic**.

### Development

#### Basic

* The developers have up front knowledge before they start coding of the frameworks that we will be using
* The developers use peer mentoring
* The coders follow an agreed upon coding standard

#### Standardized

* The Developers use a framework to support web service development
* The Developers have sufficient knowledge of the latest Microsoft Technologies to be able to fully utilize the investments in developer tools
* The Developers use the Secure Development Lifecycle

#### Advanced

* The Developers use a framework to abstract presentation work away from business logic
* The Developers use a framework to assist with management of Identity
* Developers use a state machine engine framework

#### Dynamic

* The Developers have mandated training of the latest development technologies
* The Developers have Lead Developers with clearly assigned knowledge areas
* The Developers use Code Contracts and Test driven development

#### HIS Scores:

With a summary total of **20**, HIS can be considered **Standardized**.

#### IPD Scores:

With a summary total of **10**, IPD can be considered **Basic**.

### Requirement Engineering & UX

#### Basic

* Broad assumptions by development team that they know what to build
* Fly by the seat of their pants
* Little or no written requirements- No use UX role defined
* UX done by coders or whoever available
* No specific tools
* Limited or no customer feedback
* Data or process driven design.
* Gut feel process, left to developer discretion
* No user documentation or non-standardized, on- demand ad-hoc user documentation
* Little or no validation with stakeholders

#### Standardized

* Consistent quality and format to documented requirements
* Versioning of requirements enabled and tracked
* Requirements accessible to all stakeholders and team members- UX role defined, not necessarily staffed by a UX expert
* Some non-integrated tools for UI modeling and prototyping
* Manual customer review cycle
* User center design principals understood but supported by disconnected tools
* Some consistent user documentation

#### Advanced

* Multiple types of requirements captured
* Tracking of requirements relationships and task traceability
* Use of fully integrated tools for traceability using requirements coverage analysis reports- UX Experts involved early in project
* UI prototyping formally included in development process and integrated with IDE
* Use of tooling to support UI modeling templates
* Integrated tools produce UI/requirement/task matrix
* User centered design tool (storyboarding) integrated with requirement and task work items
* User documentation specialists

#### Dynamic

* Institution of Product Change Control Board
* Use of impact analysis reports for change requests
* Published metrics on new requirements, requirements implemented, requirements tested, and requirement change requests- UX Experts incorporate latest and greatest UI principals
* Continual improvement of UX for subsequent use
* UX designers have good understanding of technology and limitations
* Designers able to understand the intersection of ease of implementation vs. great looking UI

#### HIS Scores:

With a summary total of **30**, HIS can be considered **Advanced**.

#### IPD Scores:

With a summary total of **20**, IPD can be considered **Standardized**.