

Using Team Foundation Service in a Software Engineering Course

Michael Haugrud
Department of Computer Science and Information Systems
Minnesota State University Moorhead
Moorhead, MN 56563
haugrudm@mnstate.edu

Abstract

Important outcomes of an undergraduate level software engineering course include the employment of a software life-cycle model, working in teams, and the utilization of a version control tool. Microsoft Team Foundation Service (TFS) supports all these topics. I present two assignments designed to familiarize students with TFS so they can effectively use it to develop a term project. Furthermore, I demonstrate how to utilize TFS to monitor individual team member participation in the development process.

To be successful in the workplace after graduation, Computer Science students must learn many skills. Chief among these skills is working on a team to produce a product. Any contemporary product is much too large to be developed by an individual. Many artifacts must be created, corrected, and extended as the product is produced. These are topics that rarely are covered in lower level courses since students must first develop technical skills of problem solving and fluency in a programming language. Once these abilities have been cultivated, the focus can shift to broader software life-cycle and effective team management issues.

At Minnesota State University Moorhead, we introduce the concepts and utilize tools for life-cycle and team management in our Software Engineering course. These tools and techniques can be presented in a purely academic fashion or employed in the practical development of a project. I choose to do the latter in order to better prepare students for future employment opportunities. But how can students be encouraged to put these concepts into practice? Microsoft Team Foundation Service (TFS) is an answer.

TFS can be freely used by teams of five or fewer members. I prefer to use teams of two members (with the instructor as a “silent partner”) because two students are more likely to hold each other accountable. TFS can be installed locally but Microsoft also hosts it remotely on their own servers. I prefer the latter option for simplicity’s sake. The availability of this service, as hosted by Microsoft, is generally very good. There are occasional down times but perhaps this is a valuable lesson also.

TFS provides a framework for following several agile software life-cycle models: Scrum, Agile Development, and Capability Maturity Model Integration. A storyboard with task scheduling, assignment, and tracking is available.

TFS also supports two version control utilities: Team Foundation version control and Git. Both of these integrate with Microsoft Visual Studio 2010/2012 and Eclipse. Thus students have access to a full featured development environment. Virtually any file type and programming language is supported.

In the first assignment, the students, working in teams of two, initialize their project on TFS. This entails establishing the URL for the team (Figure 1) and choosing a life-cycle model and a version control tool for their project (Figure 2). All this creates a repository for their project. From the team home (Figure 3), the project can be entered and a roster of team members created (Figure 4). They are required to add the instructor as a team member. This enables the instructor to monitor their progress, as will be described more fully below. Finally, they begin to work with the storyboard, adding tasks to it and assigning them to the team members (Figure 5).

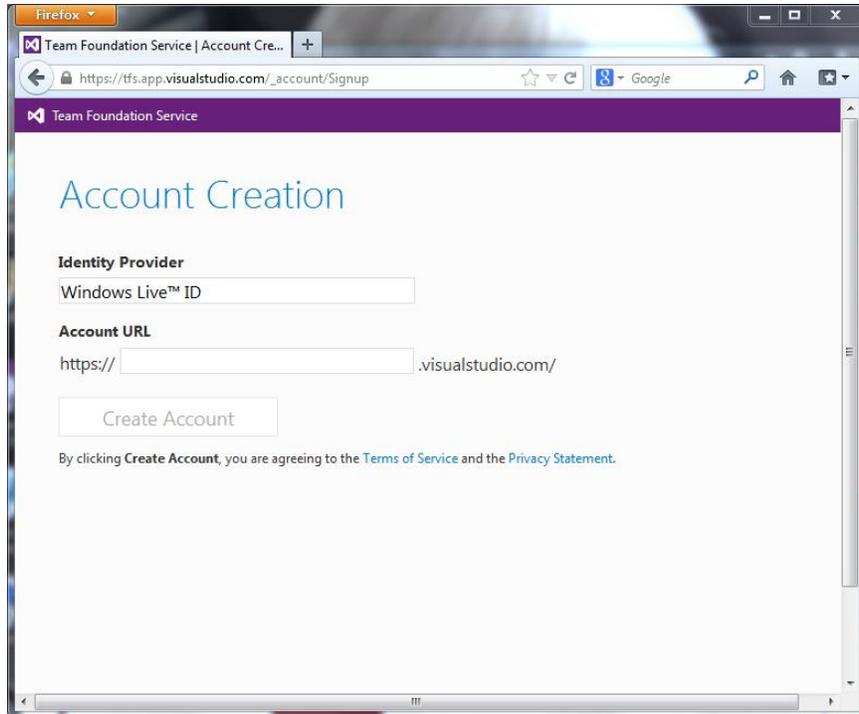


Figure 1: New TFS Account

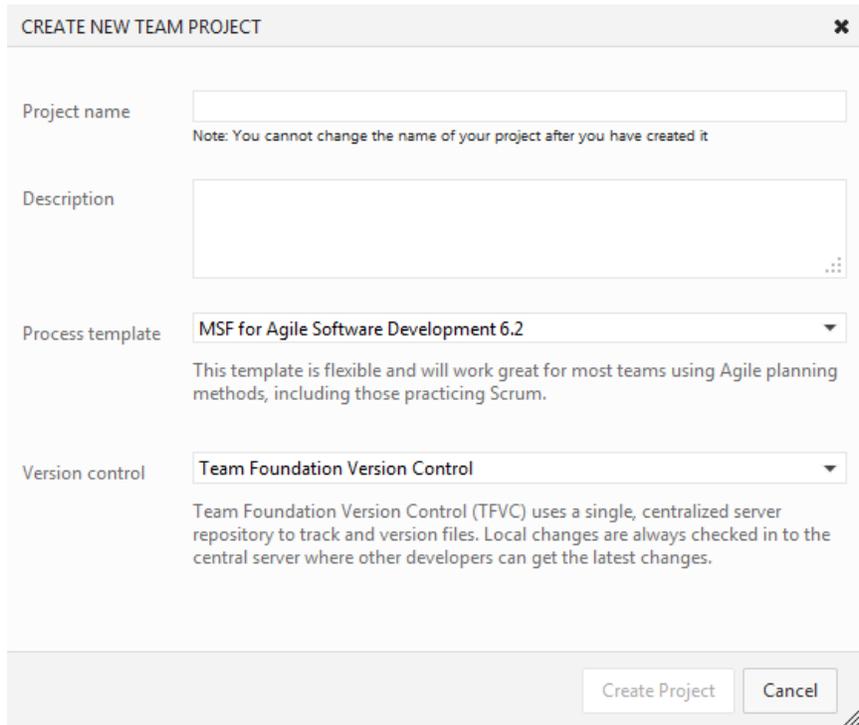


Figure 2: New TFS Project

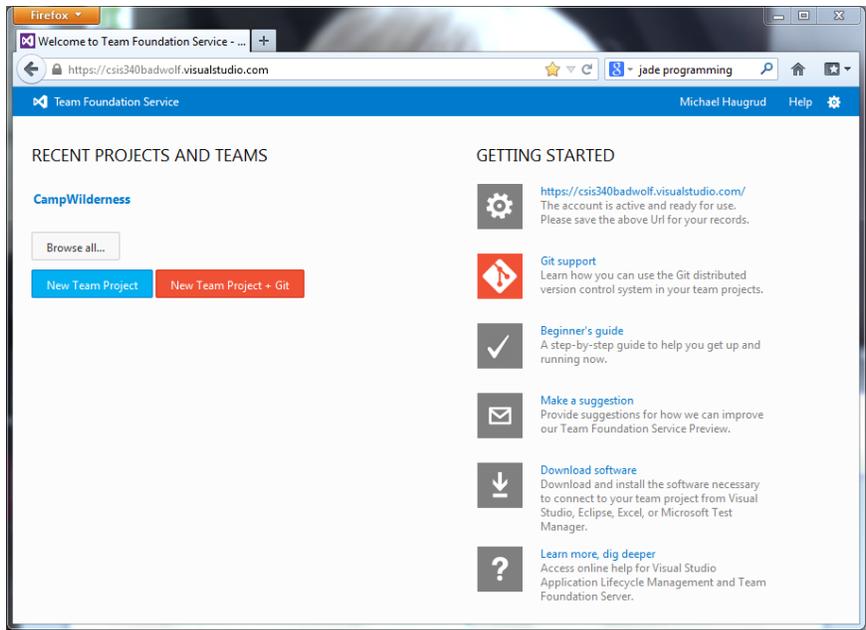


Figure 3: TFS Account Home

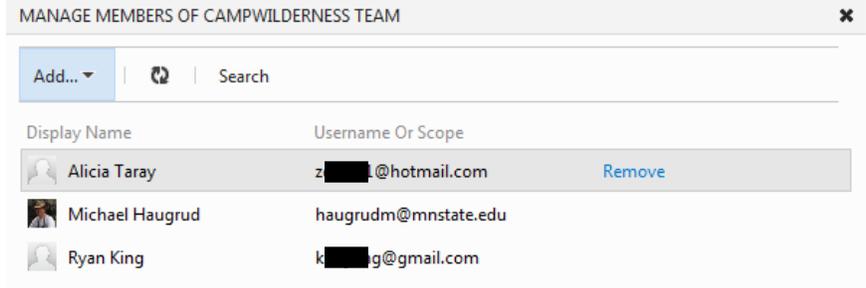


Figure 4: Team Roster

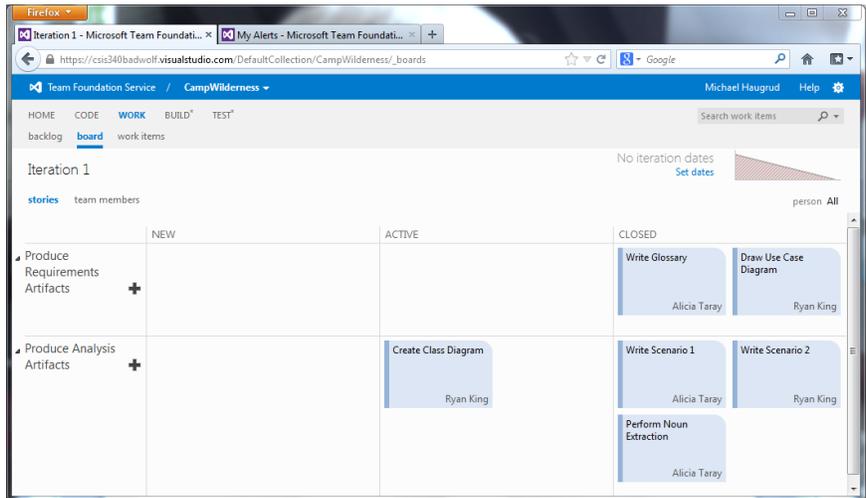


Figure 5: Storyboard

After the above assignment is complete, the instructor can log in to the project on TFS and set various alert parameters (Figure 6). This causes an email to be automatically sent to the instructor whenever the team members make any changes to the project: its storyboard or its repository.

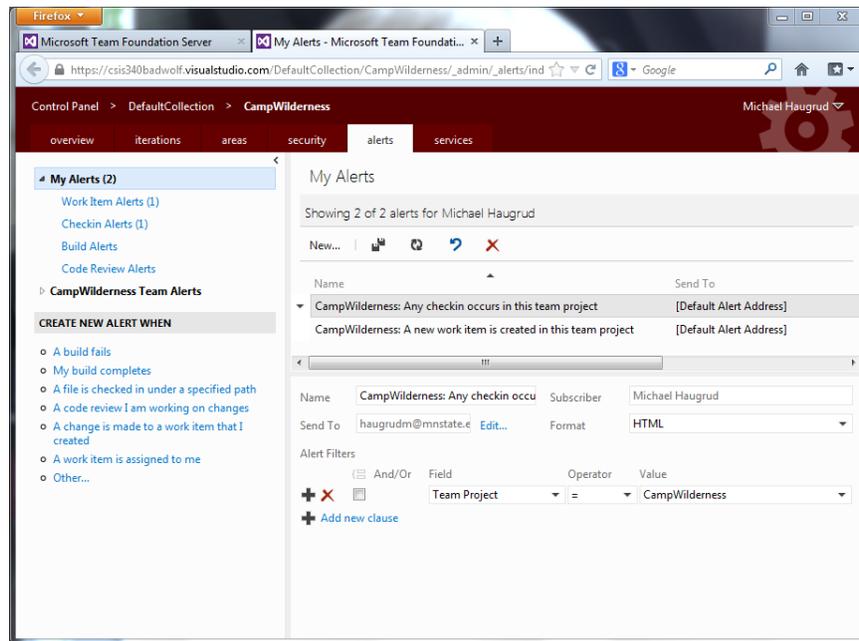


Figure 6: Manage Alerts

The email (Figure 7) shows who made the change, what the change is, and when it was done. This gives the instructor valuable insight to the level of participation by the team members.

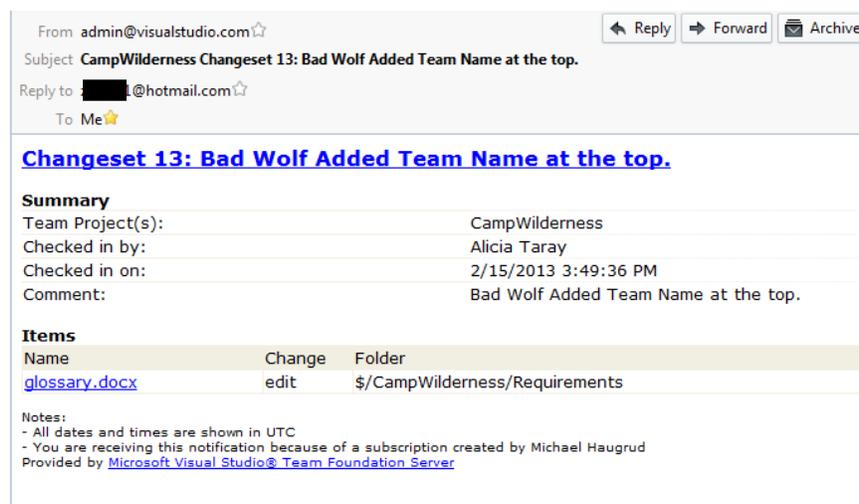


Figure 7: Alert Email

In the second assignment, the teams learn how to connect to their TFS repository and how to add and modify folders and files in the TFS repository. They begin with a simple Word document. Each team member checks out and checks in the document at least twice. This enables the students to begin to see how version control works, including some difficulties they may later encounter.

As the students change items on the TFS storyboard or repository, automatic emails are generated informing the instructor of the modifications. The instructor can then access the team's project and explore the repository (the Code section). Here one can view the history (Figure 8) of each artifact: who created it, who edited it, and when these actions were performed.

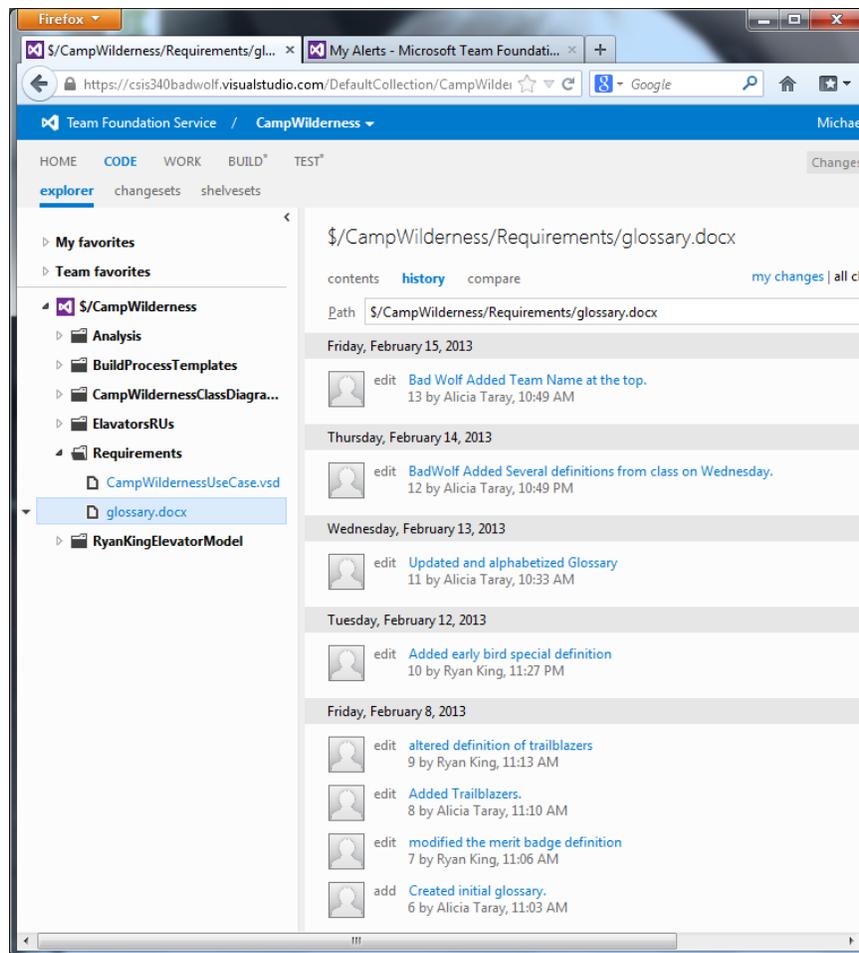


Figure 8: Revision History

By examining the change sets (Figure 9), the instructor can see exactly what was done by each student. The above capabilities allow the instructor to more accurately gauge the work done by each student and thus allocate grades more equitably.

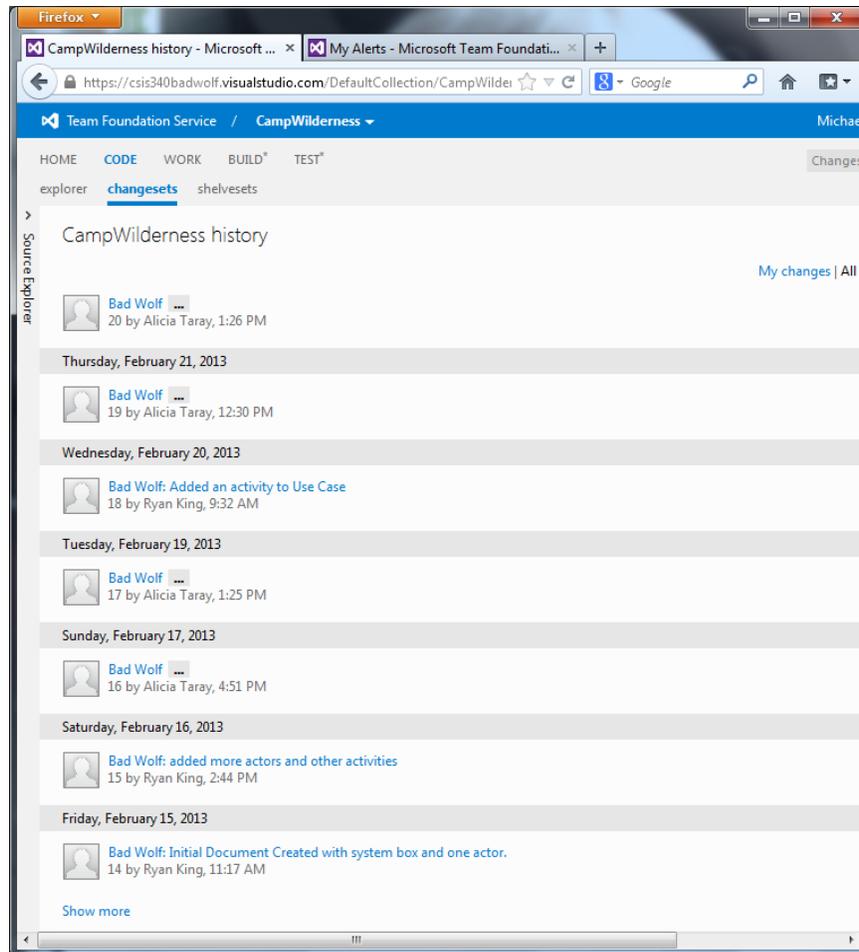


Figure 9: Changesets

In later assignments, the students will continue to update their storyboard and add more complex items to their repository. When they add Visual Studio projects, they will notice that the project consists of many files and folders, all of which must be placed under version control. They will discover many files must be checked out and checked in, even when working on a single project component.

Assignment 1

One team member will do the following:

1. Start a browser and go to tfs.visualstudio.com
2. Sign in with your Microsoft Live credentials
3. Click on Create a free account now
 - For the account URL type CSIS340yourTeamName (for example: CSIS340Dragons)
4. Click on New Team Project (blue button)
 - Project Name: CampWilderness
 - Description: you may leave this blank
 - Process Template: choose **MSF for Agile Software Development 6.2**
 - Version Control: Team Foundation Version Control
5. Navigate to Project and click on Manage all members
 - Add your partner (use his/her Live ID)
 - Add haugrudm@mnstate.edu
6. Click on your user ID in the upper right and choose My Profile
 - Display Name: enter your full name
 - Change your picture at your earliest convenience
 - Click Locale and select our time zone: Central Time (US & Canada)

The other team member will do the following:

1. Start a browser and go to <https://CSIS340yourTeamName.visualstudio.com> (the URL your partner created in step 3 above)
2. Sign in with your Microsoft Live credentials
3. Do step 6 above
4. Navigate to your team's CampWilderness project
5. Add a new User Story for Iteration 1: "Produce Requirements Artifacts"
6. View Board and add the following tasks to Produce Requirements Artifacts
 - "Write Glossary" - Assign it to your partner
 - "Draw Use Case diagram" - Assign it to yourself

Assignment 2

Team member A will do the following:

1. Start Visual Studio 2012
2. In the Team menu select Connect to Team Foundation Server
 - Click on Servers and Add the server csis340yourTeamName.visualstudio.com (use HTTPS)
 - Sign in using your Windows Live credentials
 - Close the Add/Remove Server dialog box
 - Connect to your team project
3. In Team Explorer
 - Double click the Source Control icon to open Source Control Explorer
 - Map your project folder on the server to a local folder on your computer
 - Click on Not mapped in the center pane
 - Click ... to the right of Local folder
 - Navigate to where you would like to make your project folder (use a flash drive in lab)
 - Click Map and Yes
4. Within Source Control Explorer
 - Right click in the main pane and make a new folder named **Requirements**
 - Right click the new folder, choose Check In Pending Changes, type a comment, and Check In
5. Start Microsoft Word
 - Create a new document
 - Type in one entry for your glossary
 - Save it (with a descriptive file name) in the folder you made in step 4
 - Close Word
6. Return to the Source Control Explorer
 - Open the folder you made in step 4
 - Right click in the main pane and choose Add Items to Folder
 - Choose the file you made in step 5 and click Finish
 - Open Requirements folder, right click the file, choose Check In Pending Changes, and Check In

Team member B will do the following on a different computer:

1. Do steps 1, 2, and 3 from above
2. Within your project folder, right click and choose Get Latest Version
3. Open the Requirements folder
4. Open the glossary in Word. **Notice it is Read-Only** so close Word
5. Right click the glossary your partner made and Check Out for Edit
6. Open the glossary in Word, add another entry to it, and save the file
7. Check In Pending Changes

Team member A

1. Right click the glossary and Check Out for Edit
2. Open the glossary in Word. **Notice it does not include your partner's work** so close Word.
3. Right click the glossary, choose Get Latest Version, and Check Out for Edit
4. Open the glossary in Word, add another entry to it, and save the file

Team member B: Right click the glossary and Check Out for Edit. **Notice it is locked**

Team member A: Check in Pending Changes and close Visual Studio

Team member B

1. Right click the glossary and Check Out for Edit
2. Open the glossary in Word. **Notice it does not include your partner's work** so close Word.
3. Right click the glossary, choose Get Latest Version, and Check Out for Edit
4. Open the glossary in Word, add another entry to it, and save the file
5. Check In Pending Changes
6. Close Visual Studio
7. Using a browser go to csis340yourTeamName.visualstudio.com
 - Sign in using your Windows Live credentials
 - Explore your project
 - Click Source and View your team folders and files
 - View History (you should see who created and edited the selected item)
 - Use Compare to view file revisions (you should see how the files changes)
 - If History and Compare shows your activity from previous steps, you have successfully completed the assignment