

Managing Group Projects with Visual Studio Online

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 Visual Studio Online (VSO) supports all these topics. I present assignments designed to familiarize students with VSO so they can effectively use it to develop a term project. Furthermore, I demonstrate how to utilize VSO 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 Visual Studio Online (VSO) is an answer.

VSO can be freely used by teams of five or fewer members. I prefer to use teams of two or three members (with the instructor as a “silent partner”). With smaller size teams, the students are more likely to hold each other accountable.

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

VSO also supports two version control utilities: Team Foundation version control and Git. Both of these integrate with Microsoft Visual Studio 2010/2012/2013 and Eclipse. Thus students have access to a full featured development environment. Virtually any file type and programming language is supported. The version control repository can be hosted locally but Microsoft also offers it remotely hosted on its 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.

In the first assignment, the students, working in their teams, initialize their project on VSO. 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 task board, adding tasks to it and assigning them to the team members (Figure 5).

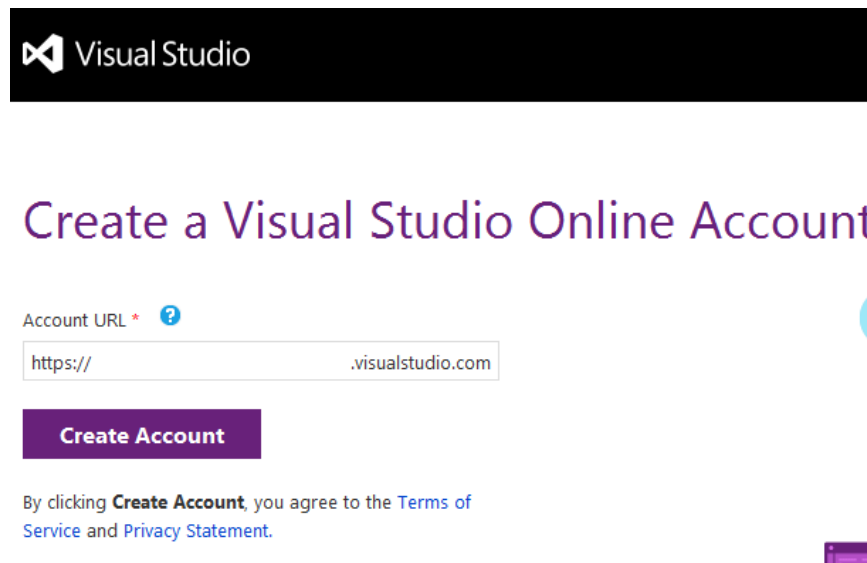


Figure 1: New VSO Account

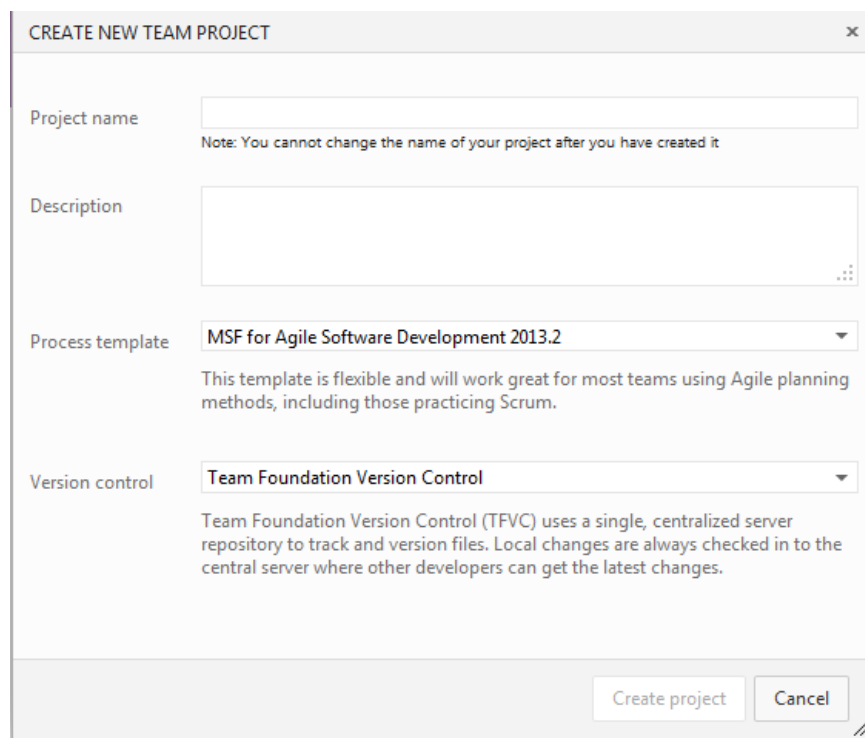


Figure 2: New VSO Project

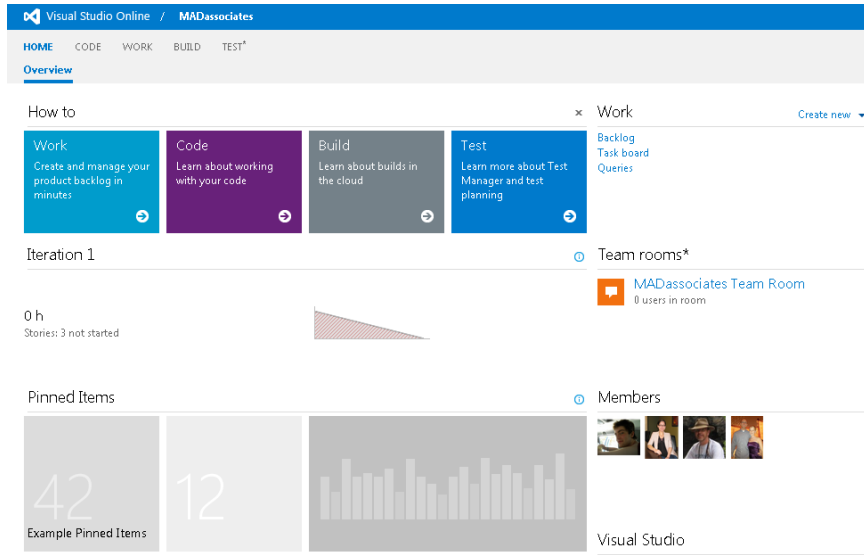


Figure 3: VSO Account Home

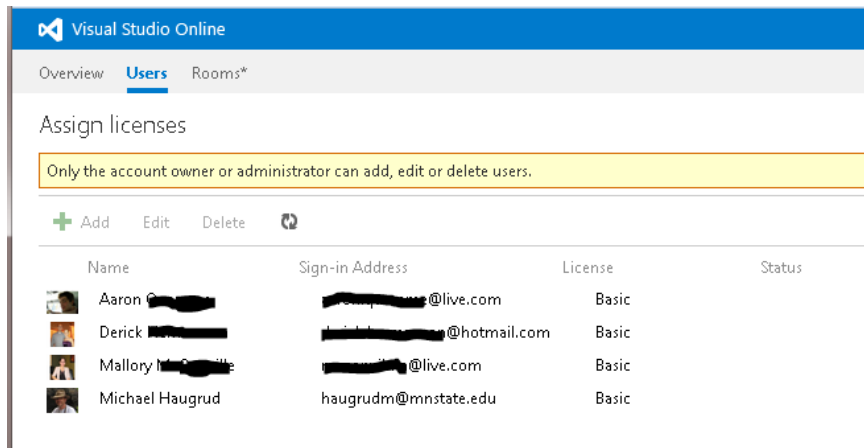


Figure 4: Team Roster

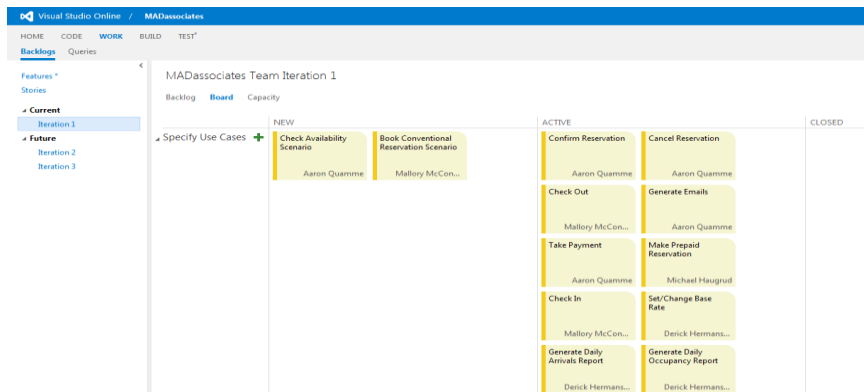


Figure 5: Task Board

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

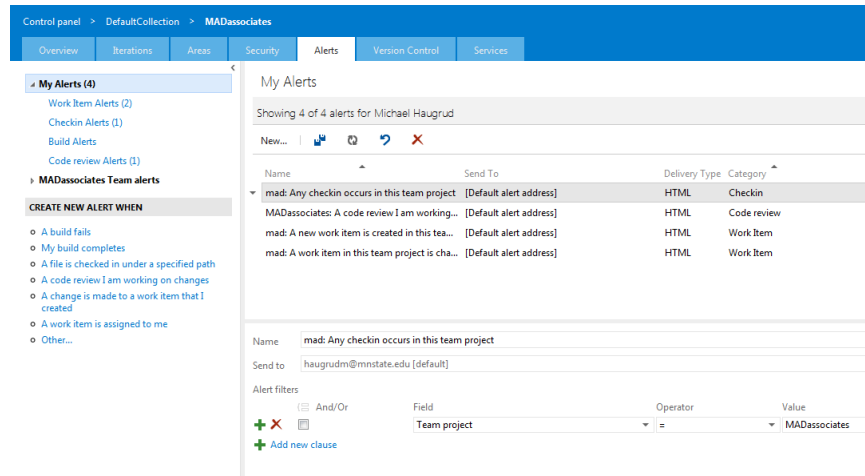


Figure 6: Manage Alerts

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

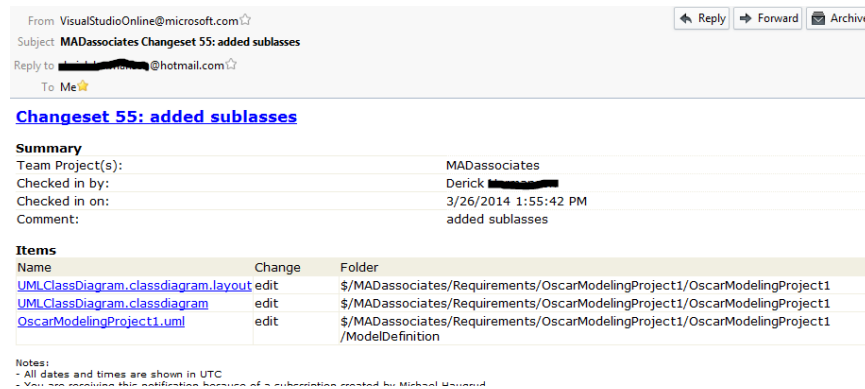


Figure 7: Alert Email

In the second assignment, the teams learn how to connect to their VSO repository and how to add and modify folders and files in the VSO repository. They begin with a simple text 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 when working with more complex content.

As the students change items on the VSO task board 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.

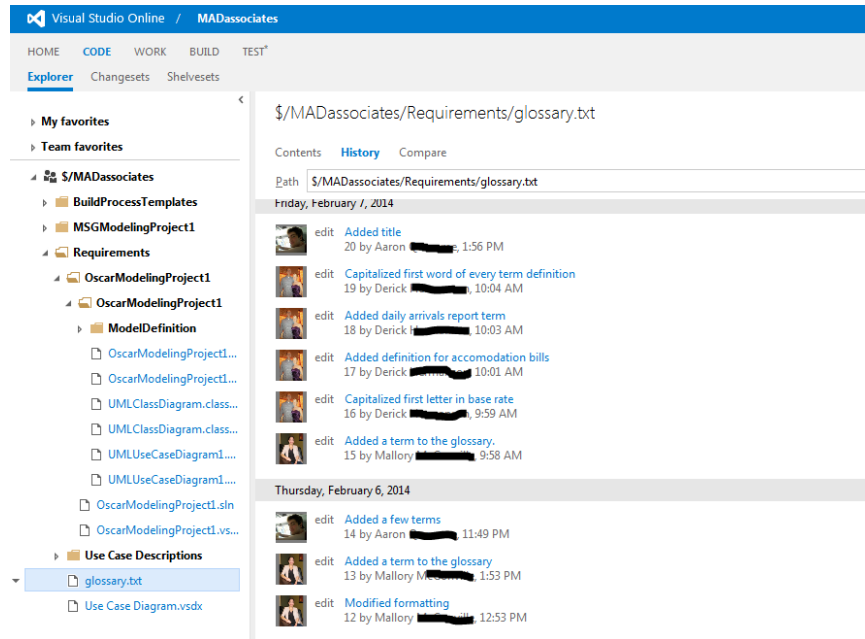


Figure 8: Revision History

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.

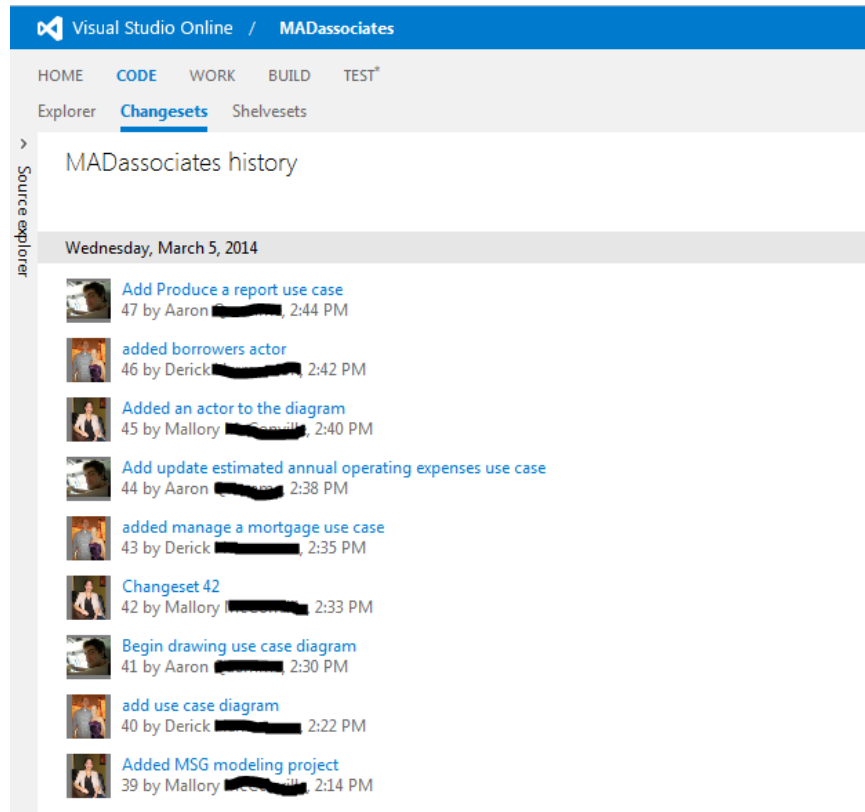


Figure 9: Changesets

In a third assignment, the students add a Visual Studio modeling project to their repository. A modeling project is considerably more complex than any of the previous items they have added. It 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

Team member A will do the following:

1. Start a browser and go to <http://www.visualstudio.com>
2. Sign in with your Microsoft Live credentials
3. Click on Create a free account now
 - For the account URL type CSIS340yourTeamName (e.g. CSIS340MSUMDragons)
4. Create a New Team Project
 - Project Name: same as your team name
 - Description: you may leave this blank
 - Version Control: Team Foundation Version Control
 - Process Template: choose **MSF for Agile Software Development 2013**
5. When the project home page is displayed click on Members Manage...
 - Add your partners (use his/her Live ID)
 - Add haugrudm@mnstate.edu
6. Click on your user name in the upper right and choose My Profile
 - Display Name: enter your full name (if necessary)
 - Change your picture at your earliest convenience
 - Click Locale and select our time zone: Central Time (US & Canada)

Team member B will do the following:

1. Start a browser and go to <https://CSIS340yourTeamName.visualstudio.com> (the URL team member A created in step 3 above)
2. Sign in with your Microsoft Live credentials
3. Do step 6 above
4. Navigate to your team's project and on its home page click on Work Create new...
5. Add a new **User Story** for Iteration 1: "Produce Requirements Artifacts"

Team member C will do the following:

1. Start a browser and go to <https://CSIS340yourTeamName.visualstudio.com> (the URL team member A created in step 3 above)
2. Sign in with your Microsoft Live credentials
3. Do step 6 above
4. Navigate to your team's project and on its home page click on Work Task board
5. Click the green plus by Produce Requirements Artifacts story and add these

Tasks

- "Write Glossary" - Assign it to team member B
- "Draw Use Case diagram" - Assign it to yourself

Team member B: move the Write Glossary task to active status on the board

All members: Study the Oscar's Outpost material on the course D2L content and begin to write your team's glossary.

Assignment 2

All team members:

- ❖ On your flash drive make a folder. Name it **Oscar**.

Team member A will do the following:

1. Start Visual Studio 2013 and sign in with your Microsoft credentials
2. Click Open from Source Control (left pane)
3. Click Select Team Projects (right – Team Explorer pane)
 - Click **Select a Team Foundation Server** and choose your team's URL
 - Click your team name project (not Oscar) and connect
4. Click Source Control Explorer (you will see your project in the left and center panes)
5. Click on **Not mapped** (upper center pane) and choose the folder you made in step ❖. Then click Yes to get a local copy of your project from the server
6. Right click in the center pane and make a new folder named **Requirements**
7. Right click your team name folder (left pane), Check In Pending Changes. In the right pane type a comment (for example, Create Requirements folder), and click Check In.

Team member B will do the following:

1. Steps 1 – 5 (same as team member A)
2. Use Windows Explorer to navigate to the folder you made in step ❖.
3. Go into the Requirements folder, right click, and create a new text document (.txt). Name it **glossary**.
4. Return to Visual Studio, double click on the Requirements folder (in center pane)
5. Right click in the center pane and choose Add Items to Folder. Choose glossary.txt and click Finish.
6. Right click your team name folder (left pane), Check In Pending Changes. In the right pane type a comment (for example, Add empty glossary), and click Check In.

Team member C will do the following:

1. Steps 1 – 5 (same as team member A)
2. Go into the Requirements folder, right click glossary.txt and choose Check Out for Edit with Lock Type of Check In.
3. Use Windows Explorer to navigate to the folder you made in step ❖ and then into the Requirements folder

4. Double click glossary.txt to start notepad and add one entry for your glossary. Save and close notepad
5. Return to Visual Studio, right click your team name folder (left pane), Check In Pending Changes. In the right pane type a comment (for example, Add first definition to glossary), and click Check In.

Team member A will do the following:

1. Right click your team name folder (left pane), Get Latest Version.
2. Go into the Requirements folder, right click glossary.txt and choose Check Out for Edit with Lock Type of Check In.
3. Use Windows Explorer to navigate to the folder you made in step ❖ and then into the Requirements folder
4. Double click glossary.txt to start notepad and add one entry for your glossary. Save and close notepad
5. Return to Visual Studio, right click your team name folder (left pane), Check In Pending Changes. In the right pane type a comment (for example, Add another definition to glossary), and click Check In.

Team member B will do the same as team member A just did

Team member C will do the same as team member B just did

All team members will do the following:

1. Use a browser to log in to your team URL and navigate to your project
2. Click on Code in the menu bar at the top (not the purple how to link)
3. Click on Contents and History in the center pane to see what has been done
4. Click on the Requirements folder and click on Contents and History
5. Click on glossary.txt and click on Contents and History
6. Click on Compare to see the difference in the various versions of glossary.txt

The previous actions happened sequentially.

See what happens when

- more than one person tries to edit the file that is **locked**.
- more than one person tries to edit the file that is **not locked**.

Assignment 3

Team member A

- Start VS2013 and login with VSO credentials
- Download your team project, map it to a local directory, get latest version
- Make a new Modeling Project (MSGModelingProject). Place it in the mapped folder under version control.
- Check in the modeling project folder (remember to comment)

Team member B

- Start VS2013 and login with VSO credentials
- Download your team project, map it to a local directory, get latest version
- Check out the modeling project folder (with lock)
- Open solution (find the .sln file in the modeling project folder)
- Add a use case diagram. Draw a use case on it.
- Save all and close solution
- Check in the modeling project folder (remember to comment)

Team member C

- Start VS2013 and login with VSO credentials
- Download your team project, map it to a local directory, get latest version
- Check out the modeling project folder (with lock)
- Open solution (find the .sln file in the modeling project folder)
- Open the use case diagram. Draw another use case on it.
- Save all and close solution
- Check in the modeling project folder (remember to comment)

Team member A

- Do the same activities as C just did (remember to get latest version)

Team member B

- Do the same activities as A just did (remember to get latest version)

Team member C

- Do the same activities as B just did (remember to get latest version)