YDM3 — YDM3 TASK 1: VERSION CONTROL USING GIT FOR GITLAB

VERSION CONTROL — D197
PRFA — YDM3

Preparation **Task Overview** Submissions Evaluation Report

COMPETENCIES

4072.1.1 : Implements Version Control
The learner implements version control processes and solutions that maintains source code.

INTRODUCTION

Version control, also referred to as revision control or source control (i.e., git), is an efficient way of managing changes to files over time. Managing files over time enables teams to collaborate freely because members can work on the same file at the same time and merge changes later. Version control and version control systems can be leveraged by anyone in an organization, from the development team to the software testing teams and even the product development/design team.

For this assessment, you will use the given scenario to create a GitLab repository and submit the link to it along with screenshots showing the work completed.

Before starting your work, review the "GitLab How-To" web link. Then, use the "GitLab" link to obtain the template. You will clone your GitLab repository to your computer to create a local repository. You will be modifying your local repository using Git Bash (Windows users) or using the Mac terminal (macOS users).

SCENARIO

You have been hired as a consultant at a company. The company has a collection of files for its website that need modification to complete the project deliverables. However, the company's contract requires that all documents be saved to a source repository, and there are other consultants working concurrently. You have been asked to make upgrades to the website.

REQUIREMENTS

Your submission must be your original work. No more than a combined total of 30% of the submission and no more than a 10% match to any one individual source can be directly quoted or closely paraphrased from sources, even if cited correctly. The similarity report that is provided when you submit your task can be used as a guide.

You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.

Tasks may ***not*** *be submitted as cloud links, such as links to Google Docs, Google Slides, OneDrive, etc., unless specified in the task requirements. All other submissions must be file types that are uploaded and submitted as attachments (e.g., .docx, .pdf, .ppt).*

Create a repository by doing the following:

> *Note: Screenshots should include the git response from the command prompt or the entire graph window.*
>
> *Note: All operations should be performed on your local machine using Git Bash or another shell unless GitLab is specifically mentioned.*
>
> *Note: You may commit and push at any time, even if a task is not complete.*

Prepare Your Repository with Initial Data on GitLab

A. Create your subgroup and project in GitLab using the provided web link and the "GitLab How-To" web link by doing the following:

1. Create a new branch named "Working" in GitLab.
2. Include a screenshot of your current repository graph in GitLab.

Clone Your Repository on Your Local Machine

B. Clone your remote repository from GitLab to your local machine using the command line interface. Include a screenshot of the command line action and be sure to have your repository name visible in the command prompt.

Make Changes, Commit, and Push

C. Modify (using any text editor) **three** HTML files on the Working branch by doing the following:

1. Commit *each* change with a short, meaningful message that explains *all* changes you have made to the **three** HTML files. Include a screenshot for *each* git command for *each* change and be sure to have your repository name visible in the command prompt.
2. Push the branch to GitLab. Include a screenshot of the command line action and be sure to have your repository name visible in the command prompt.

Create a New Branch and Make a Change

D. Modify a file on a new branch in your local repository by doing the following:

- create a "Test" branch using the command line interface
- add your student ID to the README.md file
- push the changes to the remote repository in GitLab
- include a screenshot of the command line action and be sure to have your repository name visible in the command prompt

1. Include a screenshot of the current repository graph in GitLab after pushing the changes.

Simulate a Merge Conflict

E. Introduce a merge conflict with the "Test" branch by doing the following:

- add the git version number to the README.md file on the Working branch
- merge the "Test" branch to the Working branch in your local repository

- include a screenshot that demonstrates the conflict of this merge command line action and be sure to have your repository name visible in the command prompt

1. Resolve the created conflict and push the changes to the Working branch in GitLab. Include a screenshot of the current repository graph in GitLab.

-

Tag a Branch

F. Specify a version for your local repository by doing the following:

- tag the Working branch as version V.1.0.0
- push the tag to GitLab
- include a screenshot of the command line action and be sure to have your repository name visible in the command prompt

1. Include a screenshot of the current repository graph in GitLab.

Add a Retrospective Directory to Repository

G. Add retrospective information by doing the following:

1. Create a directory on the Working branch named "retrospective".
2. Create a log.txt that will contain the output of the git log command in the retrospective directory.

   Note: Use "git log > log.txt" to redirect output to the file.

3. Create a "summary.txt" file in the retrospective directory and include the following:
   - summarize how the merge conflict in part E1 was resolved
   - describe the **three** changes that you made in part C1
4. Create **one** file (PDF or Word document) that contains *all* screenshots and comments from parts A–F and place the file in the retrospective directory.
5. Push *all* changes made in parts G1–G4 to GitLab.

Final Step

H. Include the repository link to your GitLab in the "Comments to Evaluator" section when you submit this assessment.

I. Demonstrate professional communication in the content and presentation of your submission.

File Restrictions

File name may contain only letters, numbers, spaces, and these symbols: ! - _ . * ' ( )
File size limit: 200 MB
File types allowed: doc, docx, rtf, xls, xlsx, ppt, pptx, odt, pdf, csv, txt, qt, mov, mpg, avi, mp3, wav, mp4, wma, flv, asf, mpeg, wmv, m4v, svg, tif, tiff, jpeg, jpg, gif, png, zip, rar, tar, 7z

RUBRIC

A1:CREATE BRANCH

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A branch is not created in GitLab.	A branch is created in GitLab but is not named correctly.	A branch is created in GitLab and is named correctly.

A2:REPOSITORY GRAPH SCREENSHOT

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A screenshot of the current repository graph in GitLab is not included.	The screenshot of the current repository graph in GitLab is incorrect.	The screenshot of the current repository graph in GitLab is correct.

B:CLONING REMOTE REPOSITORY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A screenshot of the cloned remote repository from GitLab is not provided.	The screenshot does not show that the remote repository from GitLab was cloned or that the command interface was used to clone the repository. Or the command prompt does not include the repository name.	The screenshot shows that the remote repository from GitLab was cloned and that the command line interface was used to clone the repository, and the command prompt includes the repository name.

C1:COMMIT WITH A MESSAGE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The screenshots do not show *any* HTML files.	Screenshots are not provided for *each* git command for *each* change. Or the screenshots do not show 1 or more changes committed to the Working branch, or 1 or more of the short, meaningful messages are incorrect. Or 1 or more of the messages do not logically explain the changes made to the 3 HTML files in part C. Or the command prompt does not include the repository name.	Screenshots are provided for *each* git command for *each* git change. *All* screenshots show *all* changes committed to the Working branch, and the short, meaningful messages are correct. *Each* of the messages logically explains the changes made to the 3 HTML files in part C. The command prompt includes the repository name.

C2:PUSH TO REMOTE

NOT EVIDENT

A screenshot of the command line actions is not included.

APPROACHING COMPETENCE

The screenshot of the command line actions does not show the push action of a branch to GitLab. Or the command prompt does not include the repository name.

COMPETENT

The screenshot shows a correct push of the branch to GitLab and the command line actions. The command prompt includes the repository name.

D:MODIFY A FILE

NOT EVIDENT

A modified file on a new branch is not provided.

APPROACHING COMPETENCE

The file on a new branch in the local repository is modified, but 1 or more of the given actions is not completed or they are completed incorrectly. Or a screenshot of the command line action is not provided. Or the command prompt does not include the repository name.

COMPETENT

The file on a new branch in the local repository is modified with *each* of the given actions correctly completed. The screenshot of the command line action is provided. The command prompt includes the repository name.

D1:PROOF OF REMOTE CHANGES

NOT EVIDENT

A screenshot of the current repository graph is not included.

APPROACHING COMPETENCE

The screenshot does not show the current repository graph in GitLab after pushing the changes.

COMPETENT

The screenshot shows the current repository graph in GitLab after pushing the changes.

E:INTRODUCE A MERGE CONFLICT

NOT EVIDENT

A merge conflict is not introduced by performing *any* of the given actions.

APPROACHING COMPETENCE

The merge conflict introduced with the "Test" branch does not perform *each* of the 3 given requirements.

COMPETENT

The merge conflict is introduced with the "Test" branch, and *each* of the 3 given actions is performed.

E1:RESOLUTION AND PROOF OF REMOTE CHANGES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A screenshot of the current repository graph is not included.	The screenshot of the current repository graph does not show that the conflict was resolved. Or the pushed changes are not reflected in the Working branch in GitLab.	The screenshot of the current repository graph in GitLab shows the resolution of the conflict. The pushed changes are reflected in the Working branch in GitLab.

F:TAG WORKING BRANCH

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A version of the local repository is not specified, or *any* of the given actions are not performed.	The version of the local repository could not be specified because 1 or more of the 3 given actions are not completed. Or the command prompt does not include the repository name.	The version of the local repository is specified, and *each* of the 3 given actions is completed. The command prompt includes the repository name.

F1:PROOF OF REMOTE CHANGES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A screenshot of the repository graph is not included.	The screenshot of the current repository graph in GitLab does not show the correct tag version label.	The screenshot of the current repository graph in GitLab has the correct tag version label.

G1:RETROSPECTIVE DIRECTORY

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A retrospective directory is not created.	The retrospective directory is not on the Working branch.	The retrospective directory is created on the Working branch.

G2:LOG.TXT FILE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A log.txt file is not created.		The log.txt file is created in the retrospective directory and con-

	The log.txt file is not created in the retrospective directory or does not contain the git log output information in the file.	tains the git log output information in the file.

G3:SUMMARY.TXT

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A summary.txt file in the retrospective directory is not created.	The summary.txt file in the retrospective directory does not summarize how the merge conflict in part E1 was resolved in the summary.txt file. Or the file does not provide a description of the 3 changes that were made in part C1.	The summary.txt file in the retrospective directory includes the summary of how the merge conflict in part E1 was resolved in the summary.txt file and includes the description of the 3 changes that were made in part C1.

G4:FILE WITH SCREENSHOTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A PDF or Word document containing screenshots is not provided or is not placed in the retrospective directory.	A PDF or Word document is placed in the retrospective directory, but the file does not include *all* the screenshots or *all* comments from parts A–F.	A PDF or Word document containing *all* screenshots and *all* comments from parts A–F is inserted in the retrospective directory.

G5:PUSH ALL CHANGES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
No changes are pushed.	Not *all* changes made in parts G1–G4 are pushed from part G to GitLab, including the retrospective directory.	*All* changes from parts G1–G4 are pushed from part G to GitLab, including the retrospective directory.

H:REPOSITORY LINK

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A repository link in the comment section is not included.	The repository link included in the "Comments to Evaluator" section is *either* broken or inac-	The correct and accurate repository link is included in the "Comments to Evaluator" section.

curate (i.e., the link does not arrive at the appropriate repository).

I:PROFESSIONAL COMMUNICATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
This submission includes professional communication errors related to spelling, grammar, punctuation, and sentence fluency. For best results, please focus on the specific Correctness errors identified by Grammarly for Education to help guide your revisions. If you need additional assistance preparing your submission, please contact your Instructor.	This submission includes professional communication errors related to spelling, grammar, punctuation, and/or sentence fluency. For best results, please focus on the specific Correctness errors identified by Grammarly for Education to help guide your revisions.	This submission demonstrates correct use of spelling, grammar, punctuation, and sentence fluency. You have demonstrated quality professional communication skills in this submission.

WEB LINKS

GitLab

GitLab How-To