Lab 2: Software Development Life Cycle

Task: Using Git to version control files.
Lab topic(s):   Software Development Life Cycle (SDLC)
Lab objective(s): Understanding of
- Introduction to the main stages of the system development life cycle.
- Understanding the importance of requirements analysis in software development.
- Identifying functional and non-functional requirements.

Development Life Cycle

The typical 7 stages of the system development life cycle (SDLC) are planning and feasibility, requirements analysis, design and prototyping, software development, system testing, implementation, and maintenance. Alternatively, the aforementioned processes are sometimes split into 5 phases of the system development life cycle: planning, analysis, design, implementation, and maintenance.

SDLC is not a methodology per se, but rather a description of the phases that a methodology should address. The most common development methodologies are Iterative, Spiral, Agile, Scrum, Kanban, V-model, Waterfall, Lean, and eXtreme Programming. The choice of methodology depends on the project and the team, as each methodology has its own strengths and weaknesses.

Git

Git is a distributed version control system (DVCS) that tracks changes in any set of computer files, usually used for coordinating work among programmers who are collaboratively developing source code during software development. It was originally authored in 2005 for the development of the Linux kernel.[1]

There are other version control tools and systems such as Mercurial, Subversion, Helix, Sapling, Bazaar, and Team Foundation Server, each with specific and general use cases. Git is the most popular and is integrated into almost every IDE out there. A comprehensive list of GUI clients is available here.

Collaboration

1. Create a GitHub account, if you do not have one already, and sign up for the GitHub Student Developer Pack.
2. Install Git and GitHub Desktop, if needed.
3. Login with your GitHub credentials using GitHub Desktop or Visual Studio Code.

Git Operations

1. Clone: to open a remote repository locally.
2. Fetch: to receive the latest changes.
3. Pull: download the latest changes to local repository.
4. Commit: local operation to upload changes to local repository
5. Push: network operation to transfer commits to remote repository

Individual Repository

1. Join the GitHub Classroom invitation to create your individual *private* repository.
2. Name your repository using the following format **section-username-firstname**, for example, **b55-dd0123456-dane**.
3. Clone [git clone] your individual repository locally using GitHub Desktop or Visual Studio Code.
4. Stage [git add] your changes.
5. Commit [git commit] your staged changes locally using a descriptive and concise message.
6. Push [git push] your changes to sync them with the remote repository.
7. Use this process to push your exercises.

Team Repository

1. Join the GitHub Classroom invitation to create/join your team *private* repository.
2. Name your team repository using the following format **team-sectionNo-teamNo**, for example, **team-B05-G02**.
3. Clone [git clone] your team repository locally using GitHub Desktop or Visual Studio Code.
4. Fetch [git fetch] to receive the latest changes.
5. Pull [git pull] to apply the changes to your working copy.
6. Stage [git add] your changes.
7. Commit [git commit) your staged changes locally using a descriptive and concise message.
8. Push [git push] your changes to sync them with the remote repository.
9. Use this process to push your milestones and project.

Lab activities:

Activity	Resources and notes	Estimated time
• Students write their name and SID (See first page on where to write these)	• First page of this document	5 minutes
• The instructor will show students how to use GitHub.	• This document • GitHub	20 minutes

• **Task 8.1:**
   • **ACCEPT** individual assignment using the link: https://classroom.github.com/a/0kMQPGxe
   • **Clone** your repository using GitHub Desktop.
   • **Create** a 01-initiation directory under your repository.
	• GitHub
	• This document.
		120 minutes

• **Create** a text file and move it under 01-initiation.
 • Use GitHub Desktop to
    o stage your changes,
    o **commit** them using a descriptive message, and
    o **push** your work.		

• **Task 8.2:**
    • Join Group assignment using the link:
 	
    https://classroom.github.com/a/AI4aIRcC
 	
    ** First member must create the team, other members will join using the team name.
 	
    • Clone your repository using GitHub Desktop.
    • Create a *Project* for Milestone-1
 	
    ** This will create 3 types of tasks, *To-do, In-progress, Done*.
 	
    • You can add milestone tasks as Issues, and change status as you progress.
    • Create a 01-milestone-1 directory under your repository.
    • Move all the related files that you have created so far under 01-milestone-1.
    • Use GitHub Desktop to
      o stage your changes,
      o commit them using a descriptive message, and push your work		

What to submit: To Blackboard, submit a screenshot of your team GitHub's repository.		