

## Lab 8: 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.<sup>1</sup>

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](#).

## Class Repository

1. Clone [git clone] the class [repository](#) using [GitHub Desktop](#) or [Visual Studio Code](#).
2. Fetch [git fetch] to receive the latest changes.
3. Pull [git pull] to apply the changes to your working copy.
4. Repeat this process periodically to keep your local repository synchronized with the remote one; at least once at the beginning of every lab session.

## 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, [b01-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-B03-G01](#).
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
<ul style="list-style-type: none"> <li>Students write their name and SID (See first page on where to write these)</li> </ul>	<ul style="list-style-type: none"> <li>First page of this document</li> </ul>	5 minutes
<ul style="list-style-type: none"> <li>The instructor will show students how to use GitHub.</li> </ul>	<ul style="list-style-type: none"> <li>This lab document</li> <li>GitHub</li> </ul>	20 minutes
<ul style="list-style-type: none"> <li><b>Task 8.1:</b> <ul style="list-style-type: none"> <li>Join individual assignment using the link: <a href="https://classroom.github.com/a/-WcyKxwN">https://classroom.github.com/a/-WcyKxwN</a></li> <li>Clone your repository using <a href="#">GitHub Desktop</a>.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>GitHub</li> <li>This document.</li> </ul>	120 minutes

<ul style="list-style-type: none"> <li>• Create a <code>01-initiation</code> directory under your repository.</li> <li>• Move all the files that you have created previously under <code>01-initiation</code>.</li> <li>• Use <code>GitHub Desktop</code> to             <ul style="list-style-type: none"> <li>○ stage your changes,</li> <li>○ commit them using a descriptive message, and</li> <li>○ push your work.</li> </ul> </li> </ul>		
<ul style="list-style-type: none"> <li>• <b><u>Task 8.2:</u></b> <ul style="list-style-type: none"> <li>• Join Group assignment using the link: <a href="https://classroom.github.com/a/i8gUp_1Q">https://classroom.github.com/a/i8gUp_1Q</a></li> <li>• ** First member must create the team, other members will join using the team name.</li> <li>• Clone your repository using <code>GitHub Desktop</code>.</li> <li>• Create a <i>Project</i> for Milestone-1</li> <li>• ** This will create 3 types of tasks, To-do, In-progress, Done.</li> <li>• You can add milestone tasks as To-dos, and change status as you progress.</li> <li>• Create a <code>01-milestone-1</code> directory under your repository.</li> <li>• Move all the related files that you have created previously under <code>01-milestone-1</code>.</li> <li>• Use <code>GitHub Desktop</code> to                     <ul style="list-style-type: none"> <li>○ stage your changes,</li> <li>○ commit them using a descriptive message, and push your work</li> </ul> </li> </ul> </li> </ul>		