PROGRAM DEVELOPMENT FOR A GRAPHICAL ENVIRONMENT

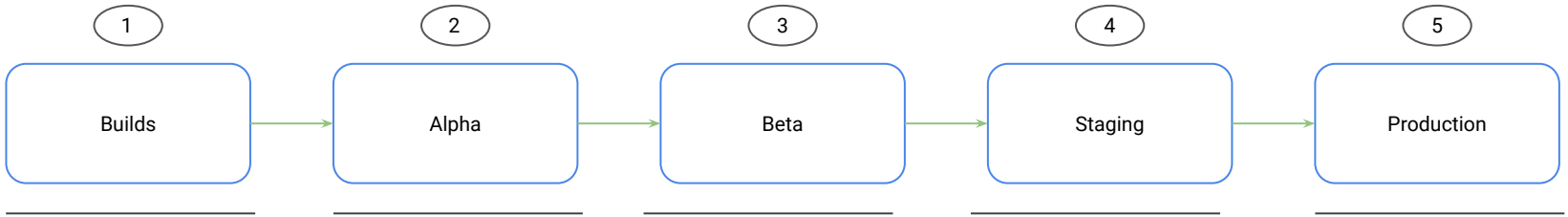
PROGRAMMING DESKTOP USER INTERFACES USING JAVA

SESSION-6

Software Versioning

Software Development Stages

A software application code base passes through multiple stages during development until the deployment of the application to be used by intended users.



Builds are either internal to the development team or used toward the release of the software.

Alpha releases are usually used by the development team to do preliminary tests, the software is expected to be incomplete and have bugs.

Beta releases are usually used by the development team and share with the clients or public for testing, they are expected to be almost complete but have some bugs.

Staging releases are expected to be complete and bug free, and are deployed in an environment as close as possible to production.

Production releases are expected to be the final versions of the software that must be complete with all features and bug free.

Software Versioning

No matter how small a change in a software code base is, it requires a new version.

- Major change indicates that a considerable modification is made to the code base. Usually, many functionalities of the software are changed or the code was redesigned. Major releases of the software do not guarantee the same behavior or backward compatibility of older versions.
- Minor change indicates that some changes are made to a code base but they are less impactful than a Major change. Minor releases usually still support the same behavior and backward compatibility as older versions but they are not guaranteed to do so.
- Patch change usually result from a fix to a flaw, bug, or malfunctioning part of the software. Patches usually do not alter the features of a software and are backward compatible.

A common way to indicate a software's version number is to use the format: `Major.Minor.Patch-Label`

A label is a textual suffix separated by a hyphen from the rest of the version number. The label itself can be any text describing the nature of the prerelease.

Examples: 1.0.2, or 1.0.2-alpha.1

More reading about the formal semantics of versioning: https://semver.org/

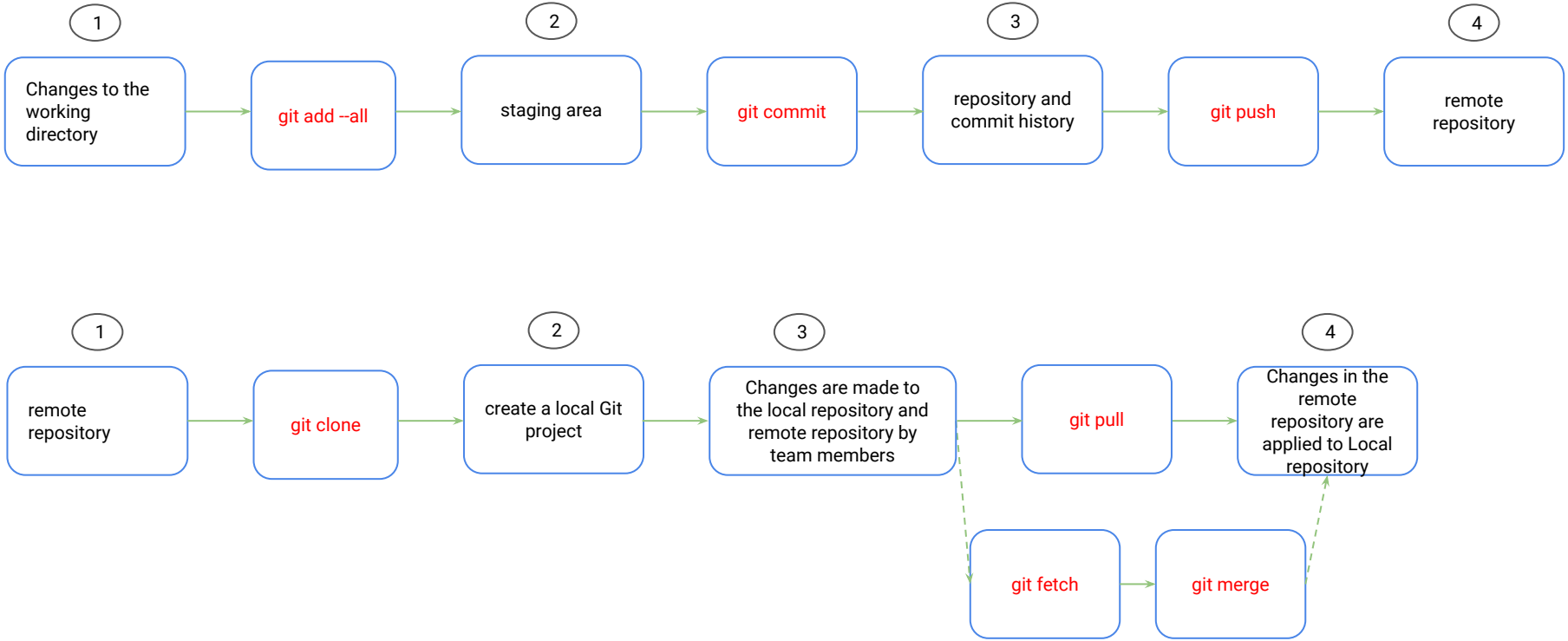
Git Repository

Git project areas

The main areas of a git project:

- Working directory
  - Where the developer works on the project's files making edits, deletions, and adding files.
  - Files in the working directory are not version controlled unless added to staging and committed. They are called untracked files.
- Staging area
  - It is the version of the project being prepared to be saved, and is represented by the "index" file.
- Commit history
  - A log of all commits. A commit is a version of the project that is saved and is associated with a unique signature called the committee hash. The commits are stored in the "objects" file inside the ".git" folder.
  - Once a file is committed it is called a tracked file.
- Local repository

Git process


1
Changes to the working directory
git add --all
2
staging area
git commit
3
repository and commit history
git push
4
remote repository
1
remote repository
git clone
2
create a local Git project
3
Changes are made to the local repository and remote repository by team members
git pull
4
Changes in the remote repository are applied to Local repository
git fetch
git merge

SUMMARY

- Software versioning
- Git repositories

Thanks!

Please contact me through Omnivox

OR

elkhawac@vanier.college