Week 1 Workshop

Getting to know your class

This week...

In this week's lab, you will:

- meet your workshop facilitators and make a note of their contact details
- meet your classmates and learn their technical interests
- begin to consider possible teammates for the project
- research the programming environments available to you
- begin to learn about Git and GitLab (if you have not already done so)

Finding your workshop

At least at the start of the semester, your workshop will be conducted online. Please check Allocate to see which class you're in, then check Unit Information in Moodle to find instructions and links.

Task 1. Meet the staff

Your facilitators should begin by introducing themselves and sharing their names and contact details. Make a note of this information in case you need to get in touch with them outside class.

Task 2. Meet your classmates

The assessment in FIT2101 is based around a major team project. You'll be making significant decisions about your project, including the language you'll be coding in and the architecture of your system. It's important to learn about the technical interests of your classmates before you choose your team.

We suggest that you do the following *as a class*:

- Facilitators will create a table in a Google Sheet with the following columns:
  - Name
  - Course
  - Computer languages known
  - Technical interests
- Facilitators will share a link to the table with the class
- Fill in your table. Each student in the class should add their own line.

  Make sure that you bookmark the sheet or add it to your own Google Drive so that you can look at it after class. This will help you later if you need to think about who you'd like to collaborate with on the project.

Note: by *technical interests*, we mean the kind of technical roles that you are interested in. Examples might include game development, quality assurance, web development, UI design, etc. You'll be able to use this information later on, when you're putting your team together and deciding on how to assign tasks. If you doesn't have any particular interests, that's okay – you can leave this column blank.

Task 3. Agile process models

Go into a breakout room with three or four fellow students.

This week's compulsory readings included a brief history of Agile process models, but does not go into details on these. We will be learning Scrum this semester, but it would be good to understand the differences between Scrum and other Agile practices.

Working with the other students in your breakout, and using any online resources you like, see what information you can find about:

- Lean software development
- XP (eXtreme Programming)
- Crystal
- the Scaled Agile Framework (SAFe)

Consider the dates these models were introduced, the people who popularized them, and whether there is evidence that they are (or have been) used in industry. If you can do so in the time available, see if you can figure out how they differ from each other, and from Scrum.

Write your information into a shared Google Document. Make sure you keep a link to it so that you can use it as an exam study resource.

Task 4. Make sure you can access the FIT GitLab server

Note: if you have only just enrolled in FIT2101, your git account might not have been created yet, it's okay to leave this task and the next one until later.

Go to `https://git.infotech.monash.edu` and log in with your Monash Authcate details. If you can't access this server, let your facilitators know or email the Chief Examiner, Robyn McNamara (`robyn.mcnamara@monash.edu`), and she will request that you be granted access.

Task 5. Basic GitLab usage

Note: This task is optional if you've used git before, but you might want to do it anyway as a refresher if you haven't used git for a long time. Next week's lab will include exercises on more advanced techniques, including branching, merging, issues, and merge requests.

- Check that git has been installed on the computer you are using. If you are using a laptop, you can install it yourself from packages available at `https://git-scm.com/downloads`. If you are using a Windows PC and don't have administrator access, you can download Git for Windows Portable from the same URL. You do not need administrator access in order to install this system on a USB drive.

  If you wish, you may also install a GUI client from `https://git-scm.com/downloads/guis`. Some clients have "standalone" or "portable" versions that you can install to a USB drive.

- Read the Git Introduction at `https://www.alexandriarepository.org/module/introduction-to-version-control-with-git/`. This will help you understand what version control systems are for, and how they work. This document walks you through the process of creating a project on`http://www.gitlab.com`. [1]

- Even if you're using git via a graphical client, you'll need to learn at least the names of the command line options. That's because all graphical clients use those names – if you don't understand them, you won't understand the options available to you. Go to `http://try.github.io` and work through the interactive git tutorial.
- Sign into the FIT git server if you haven't already done so, or if you've logged yourself out since Task 4.
- Click the **Groups** link at the top left of your screen. You'll find it in the header next to the Projects link.
- You'll see a list of the groups you're in. Click `FIT2101-2020-sandbox`. (If you don't see this in your list, tell your facilitator or email your lecturer – they will be able to add you.)
- Create a project within this group by clicking the green **New Project** button. You'll be prompted to give your project a name – note that project names use a restricted character set. It's an error to try to include any character other than letters, numbers, underscores, dashes, or full stops.
- In order to be able to access git from your own computer from your IDE or a git GUI such as GitKraken, you'll need to create an authentication token. Here's how:
  - Click on your profile icon in the top right corner of the screen, and select **Settings**.
  - On the left side of the screen, click **Access Tokens**. Give your token a name – it doesn't matter what you call it, it's just to allow you to tell your tokens apart if you have more than one.
  - Optionally, you can enter an expiry date for your token as well. If you don't choose to do that, your token will remain valid until you choose to delete it from your list of tokens.
  - Under Scopes, click the **api** checkbox; this will mean that your token gives your IDE or git GUI full access to manage your repo.
  - Click **Create personal access token** and the system will display a string of characters. Copy this to somewhere you'll be able to find it when you need it – it needs to be kept secure because it's a password, so there's no way to get gitlab to display it again.
- Clone your new repository to your local hard drive. You can do this with the `git clone` command – you can get the URL it requires by clicking the clipboard icon next to the URL on the project page in GitLab, and of course you can also select the URL and copy it in the usual way.

  Cloning the repository puts a copy on your local drive which you can then modify as you work and send back to the server when you're done. You need to learn to work this because although you can edit files and do some basic management through the web interface, it would be very inconvenient to develop your whole project that way.
- Navigate to the folder on your local drive that was created when you cloned. Use the text editor of your choice to create a file there, then add it to your repository by using the `git add` command at a command line prompt, or using the Add File option in the GUI tool of your choice. Commit your change and push it to the server (use the `git commit` and `git push` commands on your command line or in your GUI tool).
- For this step, you'll need to work with a classmate. Have them clone your repository and modify your new file, then save the change, commit it, and push to the server. Also get

[1] Note that you *won't* be using`gitlab.com` in this unit – you'll be using FIT's own git server.

them to add a file of their own to your repository. While this is happening, make similar changes to your classmate's repository.

- Pull the latest version of your repository (`git pull`) and verify that you can now see the changes that your classmate has made.

Once you're finished with the repository, you can delete it from the server if you want to. To do this, go to your project's page, click Settings in the menu to the left and select General from the menu that pops up. You'll find the link for project removal at the bottom of this page.

More information

A good place to start with GitLab is `https://docs.gitlab.com/ee/README.html#getting-started-with-gitlab`.

If you'd like more information about git, there's a handy list of learning resources at `https://help.github.com/articles/git-and-github-learning-resources/`. Note that some of these resources relate specifically to GitHub, which has a different user interface to GitLab.