Software Testing and Inspection
LAB practice
Using Git and Remote Repository

Dr. Salim Saay
Lecturer of Software Engineering
Department of Commuter Science and Information System
University of Limerick
Salim.Saay@ul.ie

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

Remote Repository: Bitbucket

- Bitbucket is a web-based version control repository hosting service owned by Atlassian, for source code and development projects that use either **Mercurial** or **Git** revision control systems.
- Bitbucket offers both commercial plans and free accounts
- You get 1 GB file storage

Remote Repository: GitHub

- GitHub, is an American multinational corporation that provides hosting for software development and **version control** using **Git**.
- It offers the **distributed version** control and source code management functionality of Git, plus its own features.
- **GitHub** has a maximum limit of 1GB

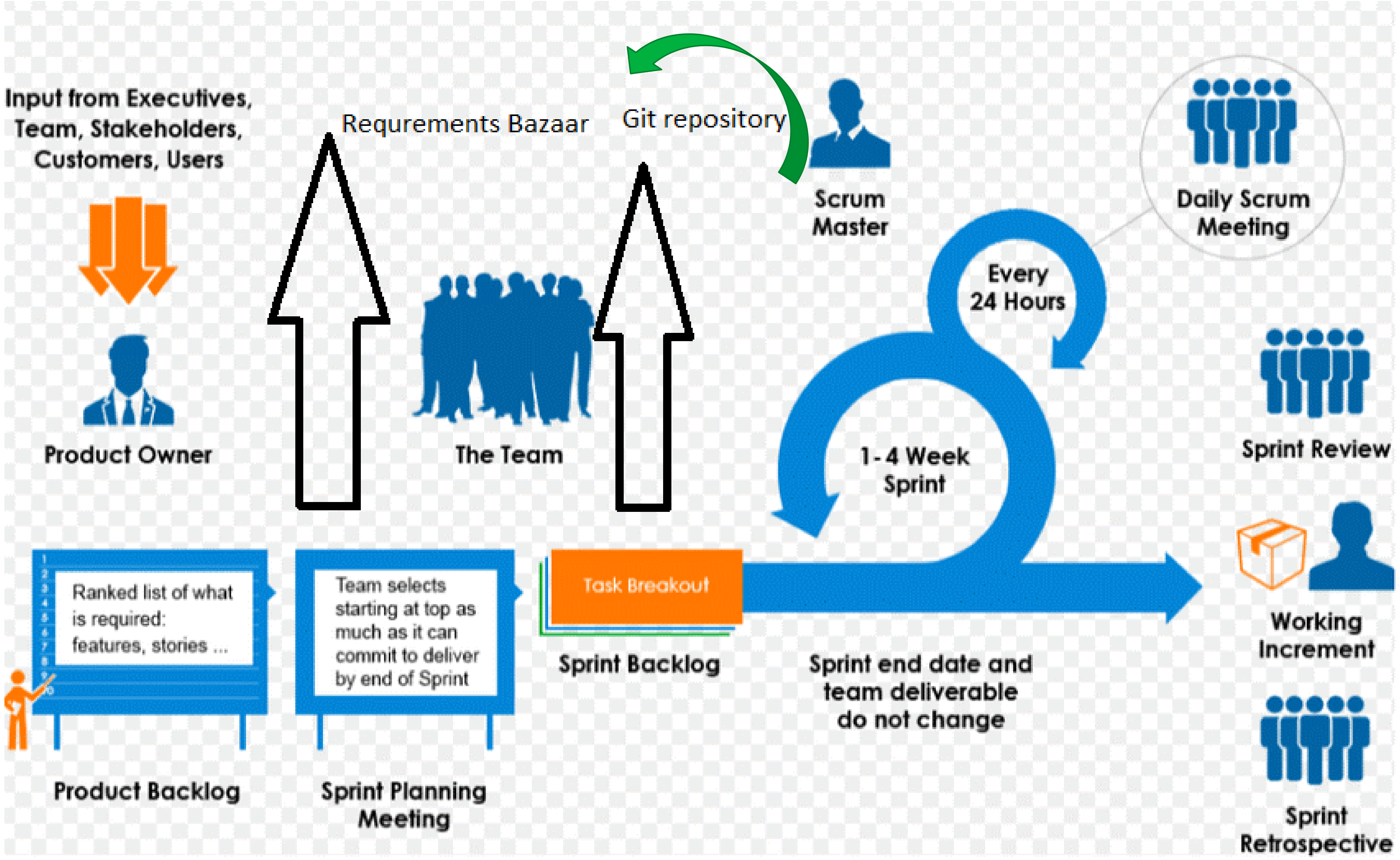
Remote Repository: JIRA

- Project management tool designed to optimize
  - **project planning,**
  - **implementation and tracking**
- Used as an issue tracking tool for software testing
- Can handle variety of issues
  - progress of a project,
  - hiring of employees,
  - product pipelines,
  - building stories for software teams, etc
- Free for teams of up to 10 users

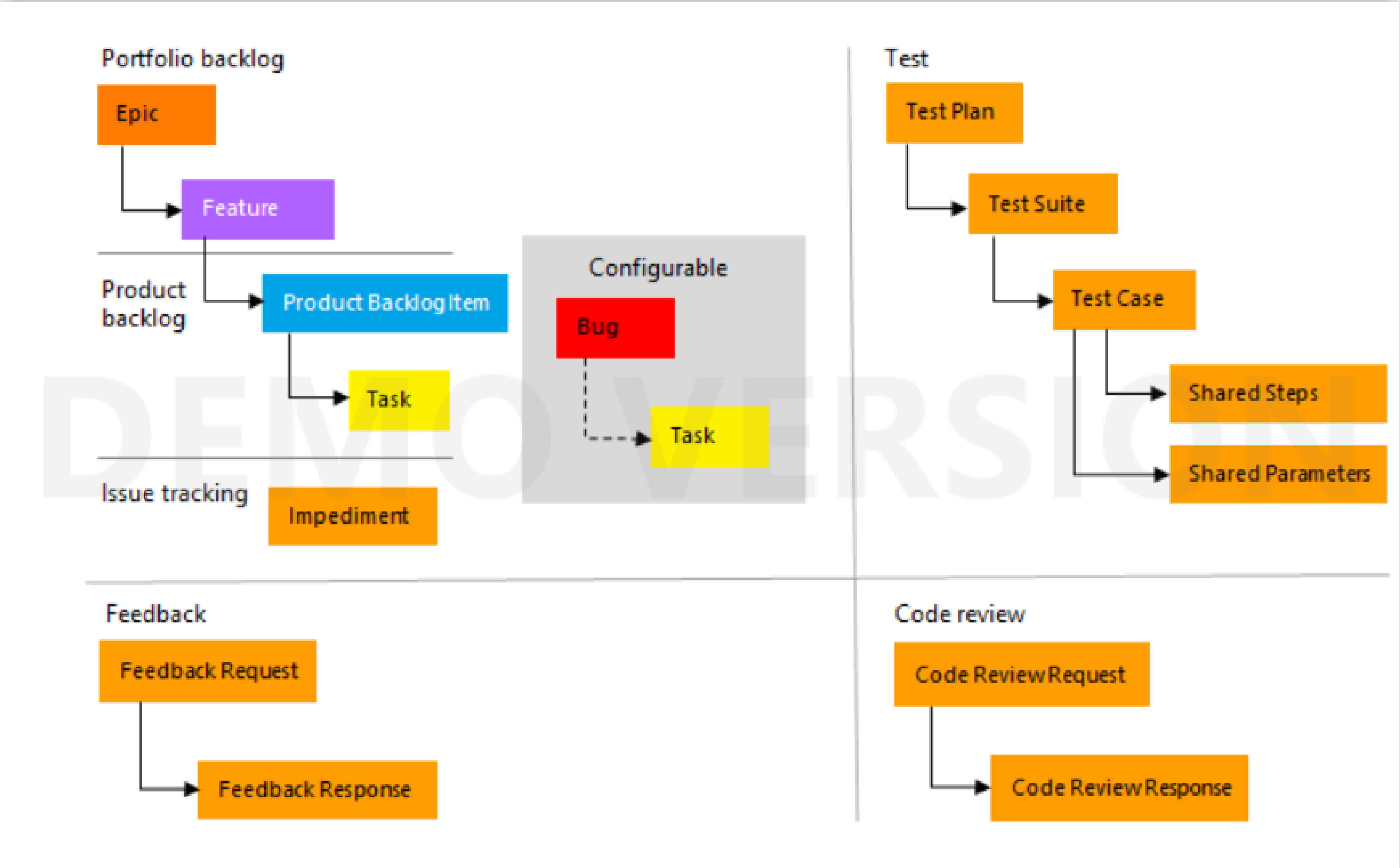
Remote Repository: JIRA

- ❖ Plan – Create user stories and issues, plan sprints, and distribute tasks across the software team
- ❖ Track – Prioritize and discuss each member’s work in full context with complete visibility
- ❖ Release – Deploy product with up-to-date information
- ❖ Report –Analyse team performance in real-time using visual data

Tools for scrum

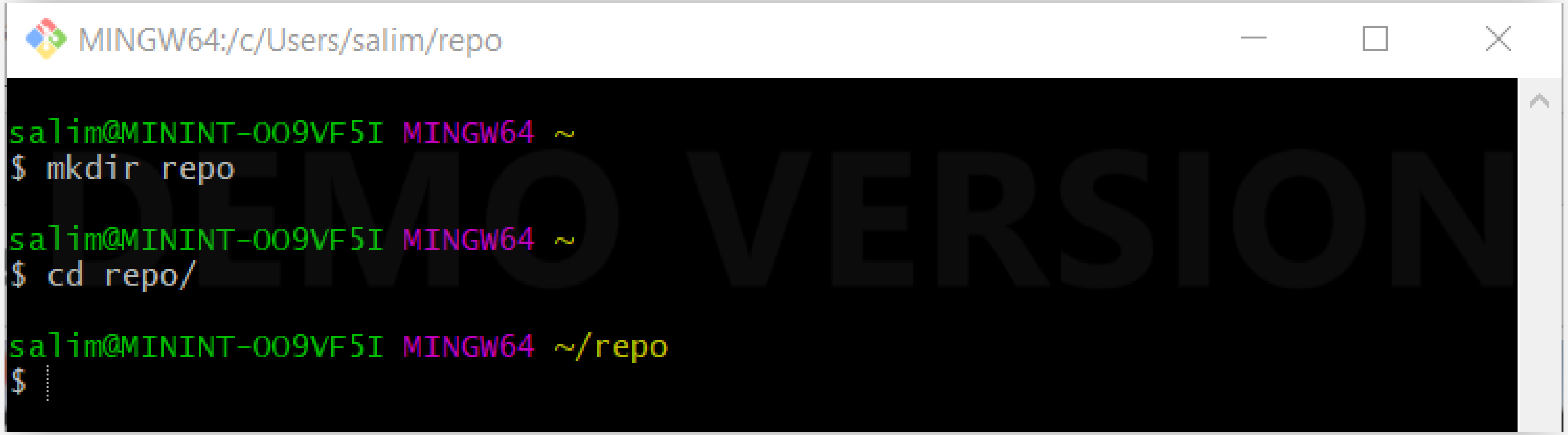

Input from Executives, Team, Stakeholders, Customers, Users
Requrements Bazaar
Git repository
Scrum Master
Daily Scrum Meeting
Every 24 Hours
Product Owner
The Team
1-4 Week Sprint
Sprint Review
Ranked list of what is required: features, stories ...
Team selects starting at top as much as it can commit to deliver by end of Sprint
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Working Increment
Product Backlog
Sprint Planning Meeting
Sprint Retrospective

Enhancing Requirements(user story) in scrum



Using Git

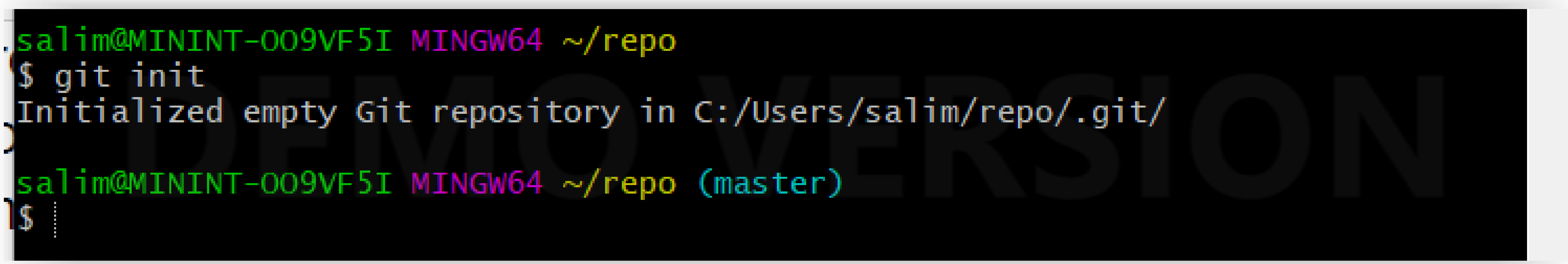
1. Install and run Git for windows https://git-scm.com/download/win
2. Create a local git repository



```
salim@MININT-OO9VF5I MINGW64 ~
$ mkdir repo

salim@MININT-OO9VF5I MINGW64 ~
$ cd repo/

salim@MININT-OO9VF5I MINGW64 ~/repo
$
```

Using Git

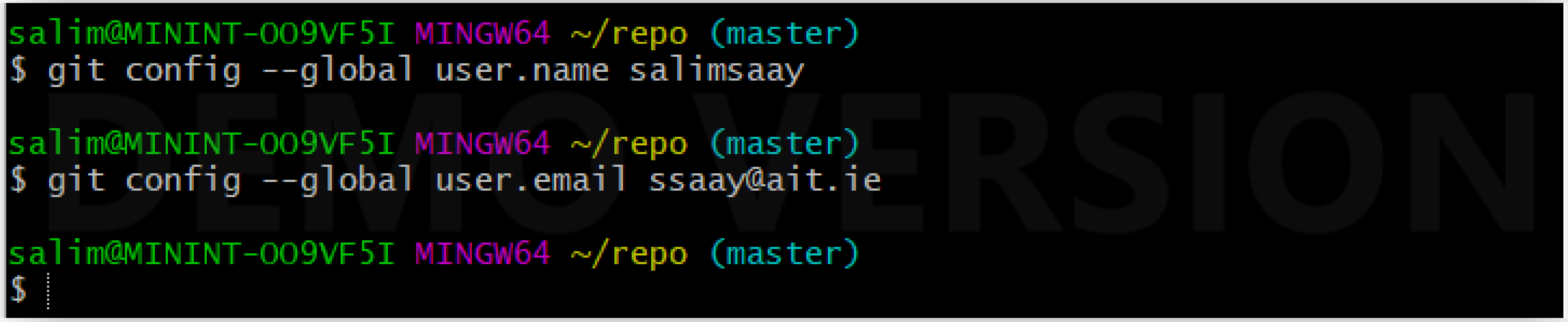
3. Initialize git repository



```
salim@MININT-OO9VF5I MINGW64 ~/repo
$ git init
Initialized empty Git repository in C:/Users/salim/repo/.git/

salim@MININT-OO9VF5I MINGW64 ~/repo (master)
$
```

Using Git

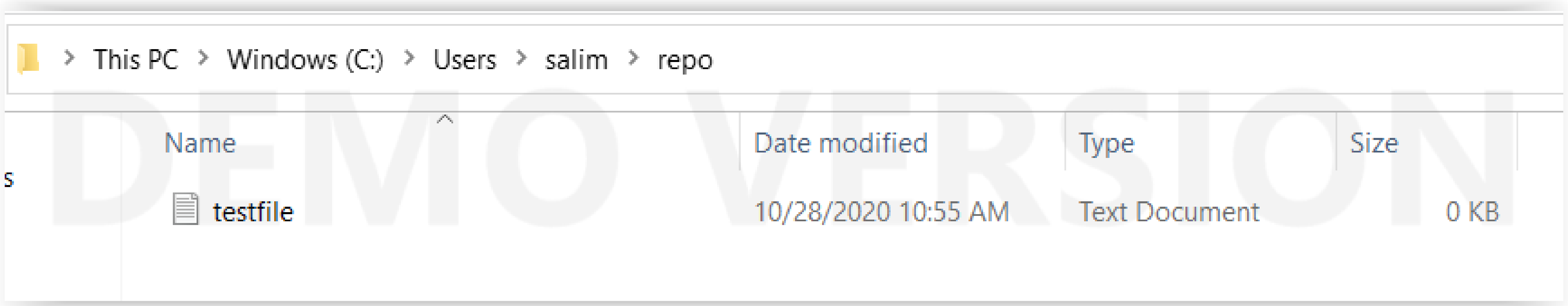
4. Add user information



```
salim@MININT-OO9VF5I MINGW64 ~/repo (master)
$ git config --global user.name salimsaay

salim@MININT-OO9VF5I MINGW64 ~/repo (master)
$ git config --global user.email ssaay@ait.ie

salim@MININT-OO9VF5I MINGW64 ~/repo (master)
$
```

Using Git

5. Add a new file to the repo



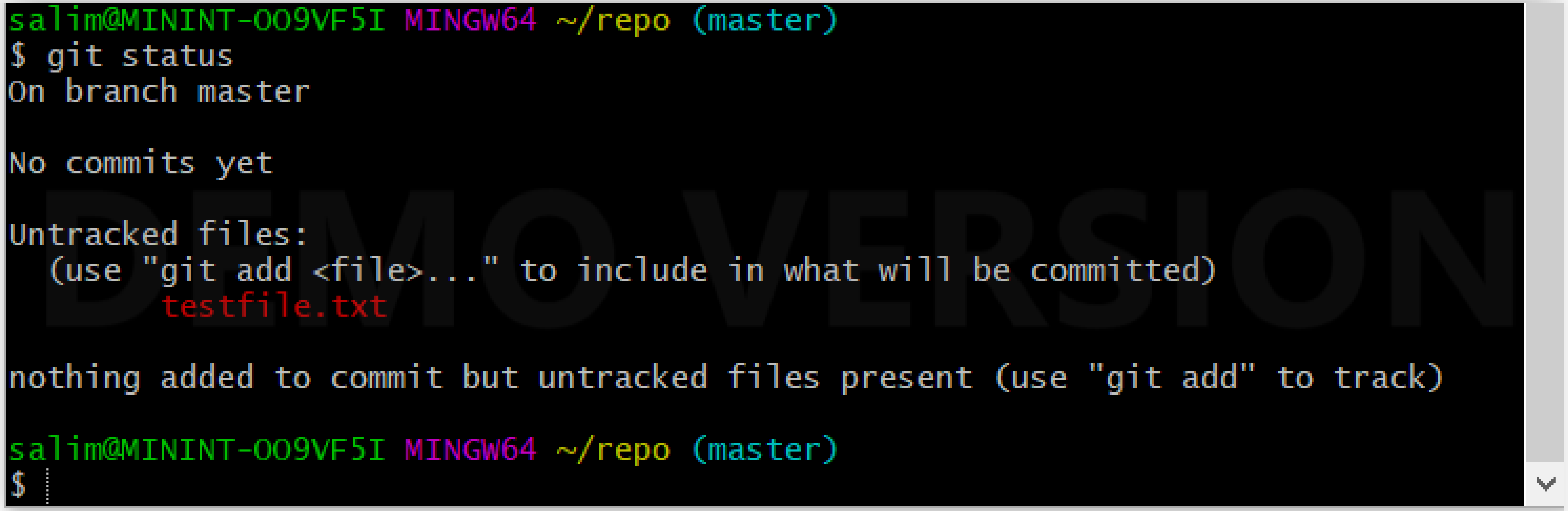
This PC > Windows (C:) > Users > salim > repo

Name	Date modified	Type	Size
testfile	10/28/2020 10:55 AM	Text Document	0 KB

Once you've added or modified files in a folder containing a git repo, git will notice that changes have been made inside the repo.

Using Git

6. After creating the new file, you can use the git status command to see which files git knows exist.

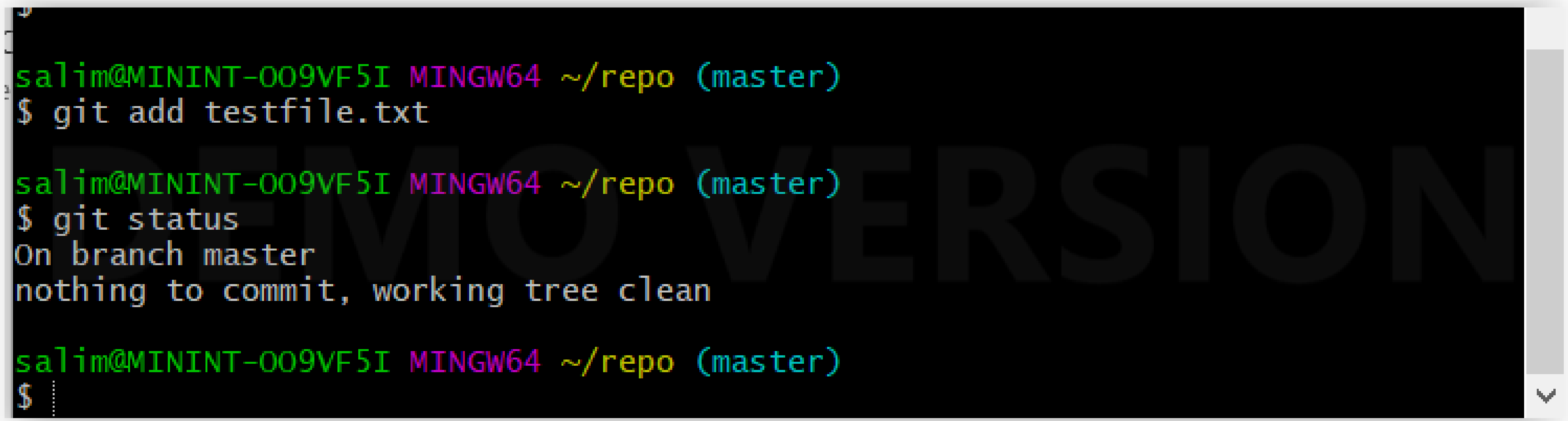


```
salim@MININT-009VF5I MINGW64 ~/repo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        testfile.txt

nothing added to commit but untracked files present (use "git add" to track)

salim@MININT-009VF5I MINGW64 ~/repo (master)
$
```

What this basically says is, "Hey, we noticed you created a new file called “testfile.txt”, but unless you use the 'git add' command we aren't going to do anything with it."

Using Git

7. Add a file to the staging environment

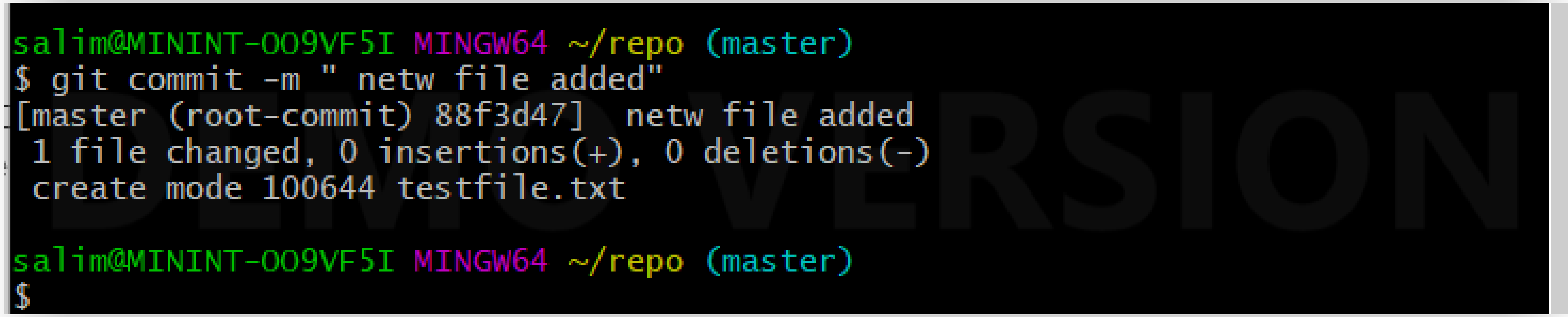


```
salim@MININT-009VF5I MINGW64 ~/repo (master)
$ git add testfile.txt

salim@MININT-009VF5I MINGW64 ~/repo (master)
$ git status
On branch master
nothing to commit, working tree clean

salim@MININT-009VF5I MINGW64 ~/repo (master)
$
```

If you rerun the git status command, you'll see that git has added the file to the staging environment (notice the "Changes to be committed" line). To reiterate, the file has not yet been added to a commit, but it's about to be.

Using Git

8. Create a commit



```
salim@MININT-OO9VF5I MINGW64 ~/repo (master)
$ git commit -m " netw file added"
[master (root-commit) 88f3d47]  netw file added
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 testfile.txt

salim@MININT-OO9VF5I MINGW64 ~/repo (master)
$
```

One of the most confusing parts when you're first learning git is the concept of the staging environment and how it relates to a commit. A commit is a record of what files you have changed since the last time you made a commit. The message at the end of the commit should be something related to what the commit contains

Remote repository

1. Open the instructions https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud
2. Create a Bitbucket account https://bitbucket.org/
3. Make a new directory and clone your git repository locally
4. Create a file and locally push it to Bitbucket
   1. Git push –-all
5. Change your file online with Bitbucket
6. Pull changes to the local repository.
   1. git pull –-all

Remote repository

1. Open the github https://github.com
2. Create an account to the github
3. Create a repository
4. clone to the git
   1. git clone https://github.com/saaysalim/AgileRepository
5. Upload the local file to github
   1. git status, git add githubtest.txt , git commit -m " thsis is the test file for github“
   2. Git Push
   3. Git pull

Software Testing: Junit

- Unit testing framework for Java.
- Unit testing – the process of checking the smallest independent
    - modules to ensure they are working as required.
- Unit testing – Manual/Automated
- Manual testing – Executing test cases manually;
- Non programmable; less reliable.
- Automated testing – Executing test cases using coding tools;
    - programmable; more reliable.
- JUnit – used for automation of unit testing.

Software Testing: Junit

- Open source framework
- Annotations and assertions – identify test methods & compare against expected results.
- Test runners
- Runs automatically and provides immediate feedback

Software Testing: Junit

- Helpful link1: www.educative.io/courses/java-unit-testing-with junit-5/B892KY261z2
- Helpful link2: www.tutorialspoint.com/junit/junit_environment_setup.htm
- Helpful link3 (Windows): www.softwaretestinghelp.com/download-and-install-junit/
- Helpful link4 (MacOS): https://stackoverflow.com/questions/21369953/need-help installing-junit-on-mac-how-to-add-junit-to-path-environmental variabl/26977630
- System Requirements – JDK and Eclipse
- Download the .jar files from this link: https://github.com/junit team/junit4/wiki/Download-and-Install
- Add the location of these .jar files to the Environment Path variables (see link3/link4).

Software Testing: Junit

❖ Open Eclipse and create a new Java project – ‘JUnitTesting’.

❖ Create a new package inside the src folder ‘<name>.junittest’.

❖ Create a new class inside the ‘<name>.junittest’ package ‘Calculator’.

```
public class Calculator {
public int add(int a, int b) {
return a + b;
}
public int subtract(int a, int b) {
return a - b;
}
}
```

Software Testing: Junit

- ❖ Right click on the project > Build Path > New Source Folder ‘test’.
- ❖ Right click on ‘Calculator.java’ in the src folder > New > JUnit Test Case > Select "New Junit Jupiter test” > Choose Source Folder as ‘test’ folder > Provide class name as ‘CalculatorTest’ > Click ‘Next’.
- ❖ Choose all methods available under ‘Calculator’.
- ❖ Add JUnit 5 Library to build path’ > ‘OK’

Software Testing: Junit

```
 class CalculatorTest {
    @Test
    public void testAdd() {
    Calculator calculator = new Calculator();
int a = 5;
int b = 4;
int actual = calculator.add(a, b);
int expected = 10;
assertEquals(expected, actual, "Sum is not correct");
}
@Test
void testSubtract() {
fail("Not yet implemented");}
}
```

Save and run ‘CalculatorTest.java’.
Look at the JUnit report to confirm if test was successful

Thank you

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH

University of Limerick,
Limerick, V94 T9PX,
Ireland.

Ollscoil Luimnigh,
Luimneach,
V94 T9PX, Éire.

+353 (0) 61 202020

ul.ie