Software Project Management - Laboratory

Lecture n° 5
A.Y. 2020-2021

Some clarification about Git

Every-time you create a file, and track it, git compresses it and stores it into its own data structure. The compressed object will have a unique name, a hash, and will be stored under the object directory.

When we commit git does:

1. If the file didn't change, git just adds the name of the compressed file (the hash) into the snapshot.

2. If the file has changed, git compresses it, stores the compressed file in the object folder. Finally it adds the name (the hash) of this compressed file into the snapshot.
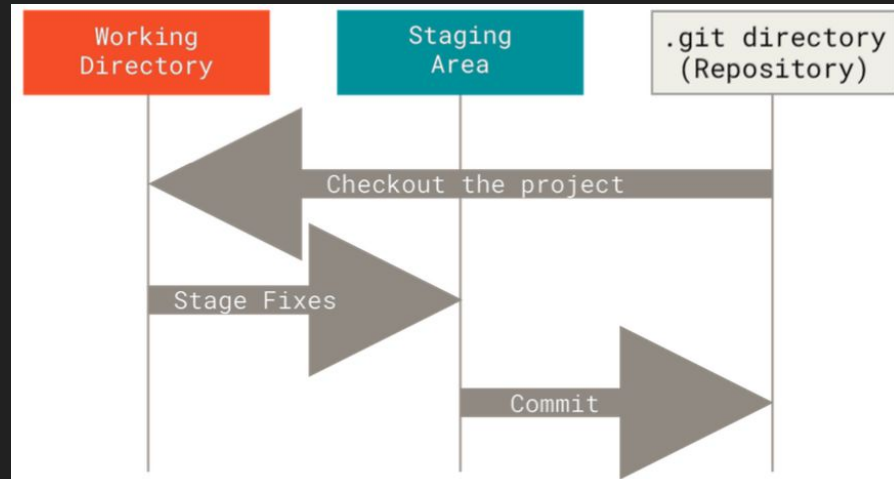

Figure 6. Working tree, staging area, and Git directory

Git Objects Status before another commit


```
[(base) fabriziounicam:MyFirsrSecondRepo user$ ls -a
.               EightFile.txt    FirstFile.txt    SecondFile.txt
..              EigthFile.txt    NewFile.txt      ThirdFile.txt
.git            FifthFile.txt    README.md
[(base) fabriziounicam:MyFirsrSecondRepo user$ cd .git/
[(base) fabriziounicam:.git user$ ls -a
.               FETCH_HEAD       config           index            objects
..              HEAD             description       info             refs
COMMIT_EDITMSG  ORIG_HEAD        hooks            logs
[(base) fabriziounicam:.git user$ cd objects/
[(base) fabriziounicam:objects user$ ls -a
.       17      45      5a      7c      a0      b0      c9      f6
..      30      49      5f      7f      a2      b9      ce      fb
00      38      4a      67      82      a3      bb      d2      fe
0a      3a      56      75      96      a8      bf      e6      info
12      3f      58      7a      99      aa      c0      f1      pack
```

Git Objects Status before another commit

```
(base) fabriziounicam:MyFirsrSecondRepo user$ vi NinethFile.txt
(base) fabriziounicam:MyFirsrSecondRepo user$ git add NinethFile.txt
(base) fabriziounicam:MyFirsrSecondRepo user$ git commit -m "I added file number nine"
[master 84c6840] I added file number nine
 1 file changed, 1 insertion(+)
 create mode 100644 NinethFile.txt
```

Git Objects Differences

Back to today's lesson...

Github Repository Setup

1. I will create a repository for each group

2. I will invite any Scrum Master to each repository

3. The Scrum Master will invite the group member as reported in the shared document

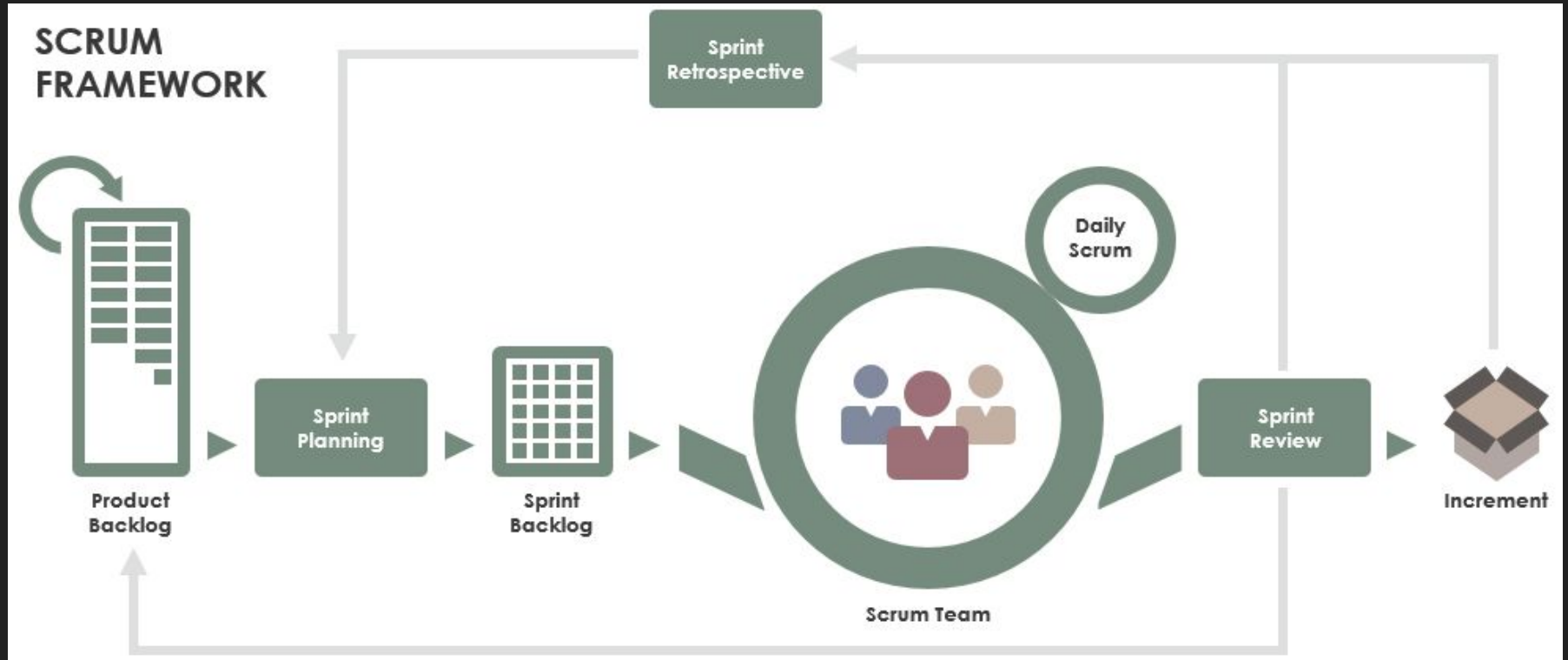4. You will start defining User Stories as Issues

SCRUM - Recap

Scrum is an Agile framework for project management that emphasizes teamwork, accountability and iterative progress toward a well-defined goal.
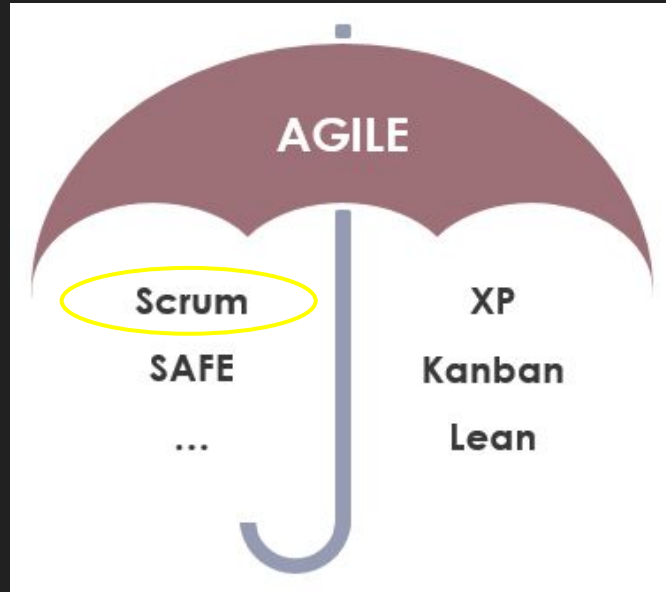
SCRUM

Roles

Product Owner - is responsible for working with the user group to determine what features will be in the product release. Some of the responsibilities:

- Develop the direction and strategy for the products and services, including the short and long-time goals;
- Provide or have access to knowledge about the product or the service;
- Understand and explain customer needs for the Development team;

Scrum Master - is the facilitator for an agile development team. Some of the responsibilities:

- Act as a coach, helping the team to follow scrum values and practices;
- Help to remove impediments and protect the team from external interferences;
- Promote a good cooperation between the team and stakeholders;

Scrum Team - is formed by 3 to 9 people who MUST fulfill all technical needs to deliver the product or the service. They will be guided directly by the Scrum Master, but they will not be directly managed. They must be self-organized, versatile, and responsible enough to complete all required tasks.

Group Management & Planning

Groups

https://docs.google.com/spreadsheets/d/1tOdMoBKzBjcHGQI3ACmrkkcaFbc2TuPfgmFQM2jL9BU/edit#gid=0

Sprint Planning

Every 2 Weeks on Friday a Sprint finishes and a new one Starts.

Friday	
Thursday	**Sprint 1**
Friday	
Thursday	
Friday	
Thursday	**Sprint 2**
Friday	
Thursday	
Friday	
Thursday	**Sprint 3**
Friday	
Thursday	
Friday	
Thursday	**Sprint 4**
Friday	
Thursday	

DevOps: Planning
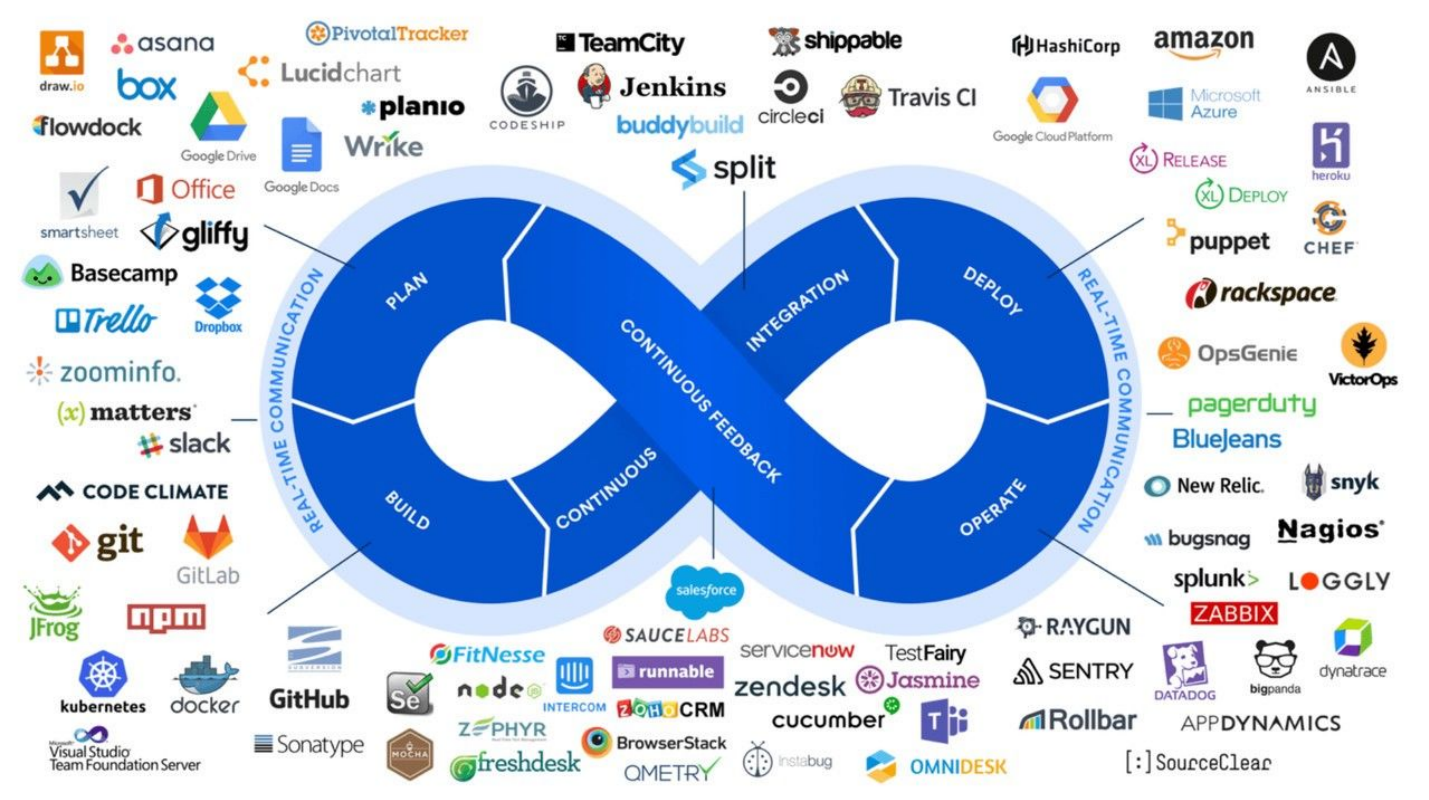
Inside Team Communication

Groups
https://docs.google.com/spreadsheets/d/1tOdMoBKzBjcHGQI3ACmrkkcaFbc2TuPfgmFQM2jL9BU/edit#gid=0

Exam Evaluation

Evaluation Schema

	Failure		Risk		Success			
Git Usage	0	1	2		3	4	4	
Testing	0	1	2		3	4	8	
SCRUM application	0	1	2		3	4	8	
Presentation	0	1	2		3	4	4	
Overall Project	0	1	2		3	4	8	
							32	32

Git Usage	2	Number of Commit	Quality of Commit	General usage (branches, contribution of group members)	
Testing	2	Numebr of Tests	Quality of Tests	Jenkins Usage	
SCRUM application/Sp	2	Capability of planning and fulfilling the planned sprint tasks; Documentation produced: User Stories, Technical Documentation (wiki)			
Presentation	2	Quality of Exam presentation: design, speaking and answers to questions			
Overall Project	2	Number of fulfilled tasks, Quality of fulfilled tasks, Presence/Absence of bugs (not detected by tests)			
	16				

You are not allowed to take a grade less than 2 and all the rest 3 or 4

Smart Parking

Actors:
- Driver
- Municipality
- Policeman
- Tow Truck
- ...

7 Total Parking Spaces
2 of 7 Spaces Available
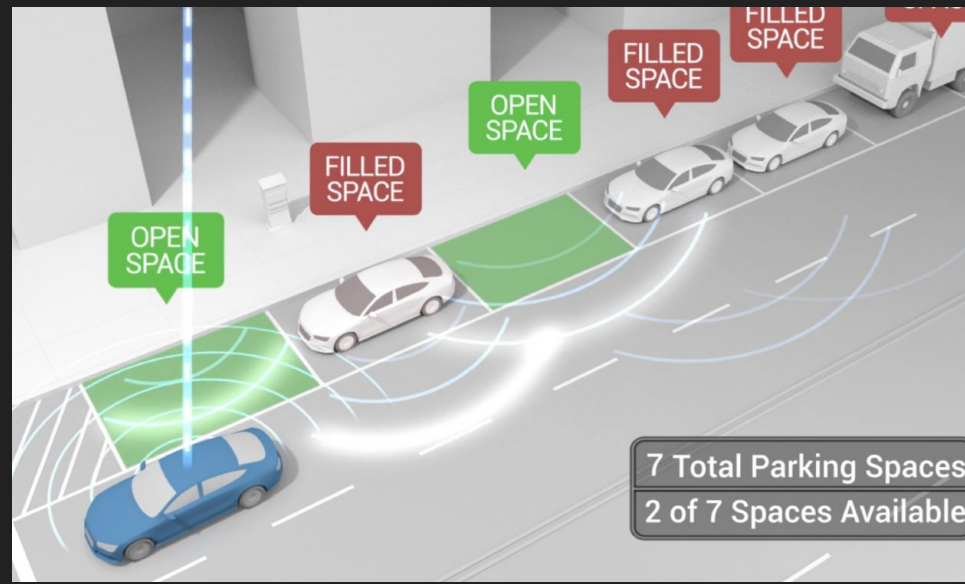
Scenarios:
- Find a Parking Spot
- Drive to the Parking Spot
- Pay the Parking Spot
- Check the Parking Spot has been paid
- Change Parking Spot Status (pay per use vs free parking spot)
- Analyse Parking Status History
- ...
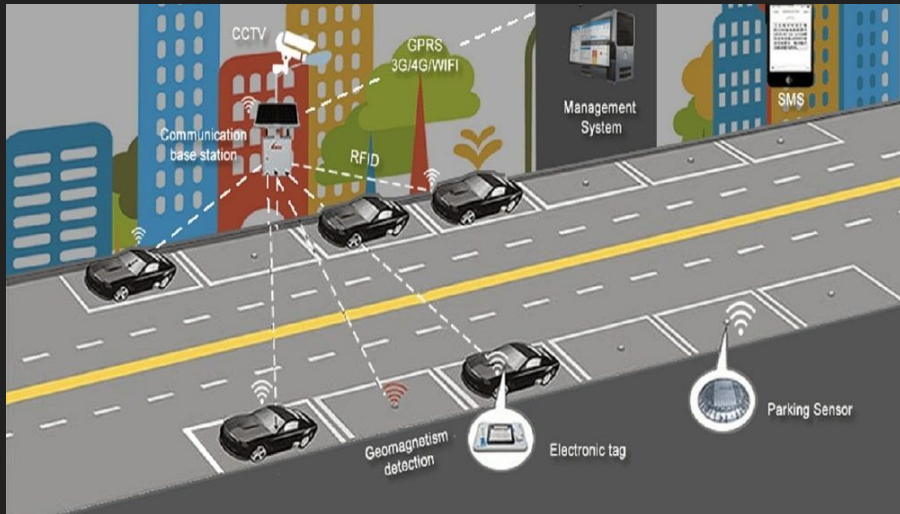
Smart Parking

Some Literature:

- Al-Turjman, F., & Malekloo, A. (2019). *Smart parking in IoT-enabled cities: A survey. Sustainable Cities and Society*, 49, 101608.

- Lin, T., Rivano, H., & Le Mouël, F. (2017). *A survey of smart parking solutions*. IEEE Transactions on Intelligent Transportation Systems, 18(12), 3229-3253.

- Saleem, Y., Sotres, P., Fricker, S., de la Torre, C. L., Crespi, N., Lee, G. M., ... & Sanchez, L. (2020). IoTRec: *The IoT Recommender for Smart Parking System*. IEEE Transactions on Emerging Topics in Computing.

- Lin, T., Rivano, H., & Le Mouël, F. (2017). *A survey of smart parking solutions*. IEEE Transactions on Intelligent Transportation Systems, 18(12), 3229-3253.

Technological Requirements

Software

- Backend in Java (e.g., Spring boot api rest)

- Git + GitHub as Version Control System

Hardware

- IoT devices which use different communication protocols (e.g. LoRa, MQTT, HTTP, etc.)

- Deploy on Cloud and/or Edge

- Possibilità di utilizzo di veri device IoT

Artifacts

The SCRUM artifacts are used to help define the workload coming into the team and currently being worked upon the team.

The main artifacts:

- Product backlog - a collection of user stories which present functionalities required/wanted by the product team. Usually the product owner takes responsible for this list.

- Sprint backlog - a collection of stories which could be included in the current sprint.

User Stories

A User Story is a simple and quick description of a specific way that the user will use the software. Generally between one and four sentences long.

Can generally follow a template:

As a *<type of user>*,
I want to *<specific action I'm taking>*
so that *<what I want to happen as a result>*.

e.g. "As a customer, I want to be able to create an account so that I can see the purchases I made in the last year to help me budget for next year."

Assign a value to estimate the effort needed to elaborate a user story (e.g., 1 to 5).

User Stories

As a *driver*

I want *to be localized on a map and search a destination*

so that *I can see the distance from the destination*

As Municipality

I want *to add a new parking space into the system*

so that *users can see and interact with the new parking space*

User Stories

As a *Team member*
I want *to search for an IoT simulator*
so that *I can abstract from the physical world for the initial phases of the project*

As a Team member
I want *to study React*
so that *we can use it for the front-end*

Github Issues

New Issues

Closing Issues

User Stories Size

Driver Login #5

⊘ Open   FabrizioFornari opened this issue 8 hours ago · 0 comments

FabrizioFornari commented 8 hours ago · edited ▾

As a driver
I want to login into the system
so that I can use it

- ☑ I want to login using facebook credentials
- ☐ I want to login using gmail credentials
- ☐ I want to login using unicam credentials