Software Engineering

Class 5

Fall 2023

Group A & B & F
Eloi Puertas - epuertas@ub.edu
Eduardo Urruticoechea - e.urruticoechea@ub.edu

Class SCHEDULE

- **[Set 20 & 22]** **Class 1: Project KickOff.**
- **[Set 27 & 29]** **Class 2: Backlog check.**
- **[Oct 4 & 6]** **Class 3: Deliver Backlog. Sprint 0 Planning.**
- *[Oct 11 & 13]* *Bank holidays*
- **[Oct 18 & 20]** **Class 4: Deliver Demo S0. Retrospective Sprint 0. Sprint 1 Planning.**
- **[Oct 25 & 27]** **Class 5: Sprint 1 check [Q1]**
- *[Nov 1 & 3]* *Examen, no class.*
- **[Nov 8 & 10]** **Class 6: Deliver Demo S1. Retrospective Sprint 1**
- **[Nov 15& 17]** **Class 7: Sprint 2 Planning [Q2]**
- **[Nov 22 & 24]** **Class 8: Sprint 2 check.**
- **[Nov 20 & Dec 01]** **Class 9: Deliver Demo S2. Retrospective Sprint 2. Sprint 3 Planning.**
- *[Dec 06 & 08]* *Bank holidays*
- **[Dec 14 & 16]** **Class 10: Deliver Final PRODUCT (S3).**

SPRINT EVALUATION

DEMO RUNNING in PRODUCTION ENVIRONMENT (not localhost !)	**Y/N**
ONE TASK COMPLETED and be part of the DEMO BY EACH TEAM MEMBER	**Y/N**
SOURCE CODE MANAGEMENT PATTERN DEFINED (Github branching strategy)	**5-10**
USING KANBAN (Github project management)	**5-10**
COHERENCE between the data in tools: Trello-Excel-Github-Demo	**5-10**
TRELLO UPDATED (Acceptance criteria)	**5-10**
WORKING Product Backlog Items at the Demo	**5-10**
Sprint template Evaluation	**5-10**

SPECIAL BONUS POINT (+1) per INDIVIDUAL CONTRIBUTION

WHY DevOps ? State of DevOps 2019 [DevOps Research and Assessment (DORA) team at Google Cloud]

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%

DevOps: Continuous Integration

➔ **Each commit should trigger a build of the software.**

➔ **Each commit should trigger a series of automated tests that provide feedback in a few minutes.**

To implement these elements, you need the following:

- **An automated build process**. The first step in CI is having an automated script that creates packages that can be deployed to any environment.
- **A suite of automated tests.** If you don't have any, start by writing a handful of unit and acceptance tests that cover the high-value functionality of your system. Make sure that the tests are reliable. That way, when they fail, you know there's a real problem, and when they pass, you're confident there are no serious problems with the system. Then ensure that all new functionality is covered by tests. Those tests should run quickly, to give developers feedback as soon as possible.
- **Run the build and automated tests on every check-in.** The system should also make the status visible to the team.

DevOps: Deployment automation

➔ **Deploy your software to testing and production environments with the push of a button**

To implement these elements, you need the following:

Inputs ready for automated deployment:

- Packages created by the continuous integration (CI) process (these packages should be deployable to any environment, including production).
- Scripts to configure the environment, deploy the packages, and perform a deployment test (sometimes known as a smoke test).
- Environment-specific configuration information.

Scripts tasks:

1. Prepare the target environment, perhaps by installing and configuring any necessary software, or by starting up a virtual host.
2. Deploy the packages.
3. Perform any deployment-related tasks such as running database migration scripts.
4. Perform any required configuration.
5. Perform a deployment test to make sure that any necessary external services are reachable, and that the system is functioning.

DEMONSTRATION

(TRELLO) SHOW User Stories and Acceptance Criteria presents at the demo	PO
(Project) SHOW Kanban status	SM
(Production) RUN working software	Team
(CI/CD) EXPLAIN: Running toolchain and deployment	DevOps
(Excel) SHOW Excel retrospective focus on the list of 4 questions	Team

FOR THE NEXT WEEK session: Sprint 1 Demo

1. End of Sprint 1. Run demo with WORKING SOFTWARE with all the classmates
2. Review Sprint 1: Run retrospective with all the classmates
3. Sprint 2 planning