Teaching DevOps at the Graduate Level: A report from Polytech Nice Sophia

Sébastien Mosser, Benjamin Benni, Philippe Collet, Guilhem Molines, Anne-Marie Déry-Pinna

Teaching DevOps at the Graduate Level
A report from Polytech Nice Sophia

Benjamin Benni[1], Philippe Collet[1], Guilhem Molines[1,2], Sébastien Mosser[1], and Anne-Marie Pinna-Déry[1]

[1] Université Côte d'Azur, CNRS, I3S, France
{benni,collet,molines,mosser,pinna}@i3s.unice.fr
[2] IBM France Lab, guilhem.molines@fr.ibm.com

Context. In this paper, we report on a course dedicated to "*N-tiers Architectures and DevOps*", taught at the graduate level at Polytech Nice Sophia since 2015. The target audience is 4^{th} year students specialized in software engineering and architecture. The course is optional, with a capacity of 50 students per year, and is scheduled on a full day for 12 consecutive weeks. It is supervised and implemented by an industrial partner who holds a part-time position in the school in addition to his daily job as a software architect. Targeted hiring companies for our students are now in the middle of a technological transformation to introduce DevOps pipelines in their organizations. It is then necessary for our students to be aware of such practices to make a difference at recruitment time.

Challenges. While creating the course syllabus, we identified the following challenges that need be tackled *w.r.t* the DevOps part of the course:

- Even if the technological stack can be hard to apprehend and deploy, tools are just a mean to an end, and the course must focus on the pillars associated to DevOps [1]: platform, deployment, testing, and people.
- Toy examples are not enough, and delivering such a content using isolated labs cannot lead to the comprehensive point of view we envisioned.

Course Description. We believe that Software Architecture and DevOps are two sides of the same coin: one needs DevOps concepts to properly implement and deliver complex architectures, and complex architectures justify such an approach. The course follows a project-based approach to support both parts and we rely on the development dimension of the project to create a continuum between architecture and operations.

On the first day, the staff delivers a *"call for bids"* that roughly describes the vision of a software product[3]. To support the development of such a system, we implemented a reference system named *The Cookie Factory* (TCF) [2]. In the first weeks, the course focuses on the concepts associated to n-tiers architectures and the pre-requisites associated to DevOps, *i.e.*, understanding modularization

[3] In the last years, we used the two following topics: *(i)* management of the slopes and lifts in a ski resort, and *(ii)* disloyalty cards granting advantages to customers who regularly shop at different places within a given neighborhood.

and testing. To support this task, students are asked to analyze the implementation of TCF (470 files, 15KLoCs). They quickly identify that it is implemented as a single monolith that needs to be modularized at all levels (business implementation, test, and deployment). This step helps them to get confidence with the project technological stack, as well as to identify why and how a DevOps approach is a good fit for such a system.

Then, based on the experience acquired with this sandbox, the product development starts. Considering the *platform* pillar, the tools selected by the students (*e.g.*, continuous integration server, testing framework, containers) must be justified and used accurately *w.r.t* the needs associated to their own project. At the *deployment* level, it is up to the students to mitigate the constraints from the development team, the operational context and the customer's expectations to create the right build plan. For the *testing* pillar, students know about unit tests and the course introduces integration and acceptance tests. Students must justify that the built product is rightly tested at these different levels. Finally, considering the *people* pillar, they have to modularize their code (and the associated tests, build plans,...) in a way that fits their development team and their business [3].

Results. The course is close to its full capacity since 2015 (137 students on 150 slots on 3 years). It is evaluated by the project delivery (code, report and oral defense), complemented by two exams (case study, 3 hours). We push students to stop being consumers of tools, and instead become DevOps architects able to identify what is necessary and how the state of practice can be assembled to support a given project. The discussions and interviews made with partners' contacts and interns' tutors are strongly positive on that point. Recruiters clearly state that such a knowledge makes a strong difference between candidates at recruitment time (interns or permanent positions). At the student level, the course received a highly positive feedback in evaluations. Student expressed as comments their surprise about the importance of the *people* pillar. We also noticed that even students who do not specialize into software architecture after the course are introducing the DevOps philosophy in their projects. From an academic research point of view, building this course also led to interesting questions about service containerization from a software engineering point of view [4].

References

1. Shaw, J.: The Four Pillars of DevOps: Agility for the Enterprise (Agile Cambridge). https://www.slideshare.net/johnfcshaw/four-pillars-of-devops-john-shaw-agile-cambridge-2014 (2014) Accessed: 2017-01-10.
2. Mosser, S.: The Cookie Factory (J2E 7 reference implementation), version 2.2. https://github.com/polytechnice-si/4A_ISA_TheCookieFactory (2017)
3. Evans, E.: Domain-Driven Design: Tacking Complexity In the Heart of Software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2003)

4. Benni, B., Mosser, S., Collet, P., Riveill, M.: Supporting Micro-services Deployment in a Safer Way: a Static Analysis and Automated Rewriting Approach. In: Symposium on Applied Computing, Pau, France (April 2018)