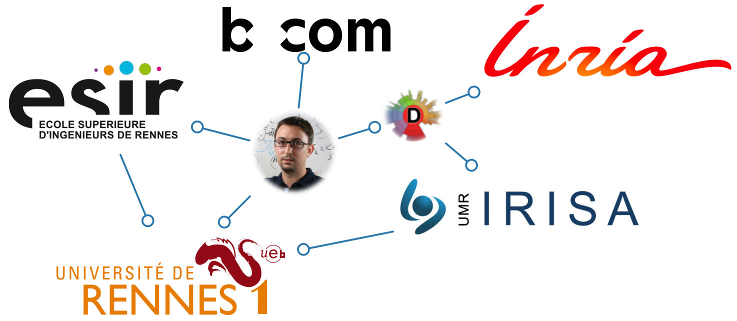
DLC – DEVOPS

OR *HOW TO BRING YOUR SOFTWARE INTO THE WILD!*

ESIR3 SI, 2020-2021

BENOIT COMBEMALE
PROFESSOR, UNIV. RENNES 1 & INRIA, FRANCE

HTTP://COMBEMALE.FR
BENOIT.COMBEMALE@IRISA.FR
@BCOMBEMALE



DLC: In Practice!

Make your own *engineering* work!

Motivation

- DevOps is large and continuously involving
- Different, complementary, topics
- Number of tools ever increasing
- Different practices for different contexts, environments, purposes…

Objective

- Technical skills
  - Apply DevOps practices through available tools
  - Apply some tools on your own application

- Soft skills
  - Explore new tools and frameworks
  - Report and synthesize on engineering practices

Organization

- Make a group of 3 and choose a topic
- Install and understand the common application (Doodle)
- Iterate
  - Explore some tools, experiment some available tutorials
  - Apply the tools on the common application
  - Document your use of the tool
- Document a synthesis of the pros and cons of the tools
- Prepare a tutorial based on the common application

- Important notes:
  - you must be proactive and autonomous (this is part of the expected skills). Communication in the team and with the teachers on Github/Teams.
  - Google, Microsoft, Amazon and GitHub offer you free stuffs.

Timeline

- Mon. Nov. 23rd: install and understanding of the common application

- Office hours

⇒ **You may validate your tools and the scenario of the final tutorial during the open hours**

⇒ **You may also use office hours to get help on the application**

- Thu. Dec. 10th (midnight, CET): push your tutorial, docs and sources as a PR on the Github repo.

Expected outcome

- **A recorded tutorial** that illustrate how to apply the DevOps tools on the common application (in between 10 and 20min): screencast, possibly with some additional slides.
- *Any documents* you created to archive what you did, you experiment, etc.
- *Any source codes* you created (especially all materials presented in the tutorial)
- **A documented Pull Request!**
- Note: speech can be in French but everything written should be in English

Topics

- Cloud-native applications, microservice architecture and orchestration frameworks
- Continuous integration, static analysis, test automation
- Software bots in software engineering
- Configuration Management Tools and Infrastructure as Code
- DevOps and Security
- Monitoring, tracing and observability
- Fuzzing, A/B and Canary Testing

DLC: Looking Ahead!

Make your own *research* work!

Motivation

- Academia and Industry are not in silo!
- Most of software paradigms come from academia
- Most of software experiences are reported to academia
- The scientific rigor is expected/appreciated by both academia and industry
  - Side note: PhD is increasingly recognized in France (and since a long time abroad)
- A software engineer must build her own, sound and strong vision

Objective

- Technical skills
  - Learn some software foundations and industrial practices
  - Step back on software technologies and build your own vision
    - *Think outside the box!*

- Soft skills
  - Discover scientific literature
    - *A first date with scientific literature?*
  - Report and synthesize a state of the art/practice

Organization

- Make a group of 3 (same than tuto) and choose an article
- Iterate
  - Read the reference(s), and make your own path from related work
  - Brainstorm with your colleagues to confront your vision on the contribution
  - Extract the essence of the contribution and make your own (group) analysis
  - Define pro and cons, expected and concrete impact, and perspectives
- Important note: Let Me Google***(scholar)*** It For You ☺
  - https://scholar.google.fr
  - But also: https://arxiv.org, https://hal.inria.fr, website of researchers and labs…
  - If eventually you have trouble to get a scientific paper, send me the DOI (or link of the paywall)

Timeline

- Mon. Nov. 23rd: choice of the article

- Office hours

  ⇒ **You may validate your presentation outline during the open hours**

  ⇒ **You may also use open hours to discuss your key findings**

- Thu. Dec. 10th (16h-18h): presentation (and will serve for the evaluation)

Presentation / Evaluation

- A presentation of 8min (+2min discussion) per group, reporting on the topic addressed
  - Speech can be in French but the slides must be in English
  - All members of the group may present a part of the presentation, but no more than 1 switch per person

- Tentative outline in 5 main parts:
  - **Context**: set the ground of the scientific/technological/industrial field
  - **Problem**: explain the problem, and position the proposed contribution
  - **Contribution**: may be a general approach, an architecture overview…
  - **Results**: may include an evaluation, discussion about pros and cons, reported impact…
  - **Conclusion and Perspectives**: give your own summary of the contribution, and give (your own) perspective

Articles

- Software Engineering at Google, by Fergus Henderson, 2019.
- How We Build Code at Netflix. M. McGarr, E. Bukoski, and B. Moyles. 2016.
- Why are many business instilling a DevOps culture into their organization?, by Diaz, J., López-Fernández, D., Perez, J., and González-Prieto, Á., 2020.
- A Large-Scale Study of Programming Languages and Code Quality in GitHub, by Baishakhi Ray, Daryl Posnett, Premkumar Devanbu, Vladimir Filkov. In Communications of the ACM, Vol. 60 No. 10, Pages 91-100, Oct. 2017.
- State of Mutation Testing at Google, by Goran Petrovic and Marko Ivankovic. In Proceedings of the 40th International Conference on Software Engineering 2018 (SEIP).
- Understanding flaky tests: the developer's perspective. Moritz Eck, Fabio Palomba, Marco Castelluccio, and Alberto Bacchelli. 2019. In ESEC/FSE 2019.
- Surviving Software Dependencies, by Russ Cox, Communications of the ACM, September 2019.
- Chaos Engineering, by Ali Basiri, Niosha Behnam, Ruud de Rooij, Lorin Hochstein, Luke Kosewski, Justin Reynolds, Casey Rosenthal. In IEEE Software vol. 33 no. 3, p. 35-41, 2016.
- A survey of techniques for approximate computing, by Mittal, S. ACM Computing Surveys (CSUR), 48(4), 62 (2016).

- Groups: https://forms.gle/L2bzTgBKzTeZG44t9
- Github repo: https://github.com/selabs-ur1/devops
- Retrieve the introduction: http://combemale.fr/course/esir/esir3/
- Common application: https://github.com/barais/doodlestudent
  - Checkout also https://hackmd.diverse-team.fr/s/SJqu5DjSD :)