Benjamin Danglot

970 Chemin de Fontmerle, Residences l'Estagnol
Les Pervenches, C, 06600 Antibes, FRANCE
0674818025
bdanglot@gmail.com
https://danglotb.github.io/
27 years old

Education

Dec2016–Dec2019 **PhD**, *Université Lille*, Villeneuve d'Ascq, Spirals team, INRIA.
Conception, implementation and evaluation of new approaches to improve test suites in DevOps context and continuous integration. This thesis takes place in the European STAMP project with 4 academics partners and 6 industrial partners.
See `https://stamp-project.eu` for more information.

2014–2016 **Master of Science**, *Université Lille 1*, Villeneuve d'Ascq, Speciality: Complex model, algorithm and data.
Machine learning, optimization, bio-informatics, data extraction

Experiences

Dec2019–Now **Research Engineer**, *Université Côte d'Azur (UCA)*, Centre de Références: Défis du Numérique.
Development of a social network to bind together people looking for job opportunities, companies looking for specific profiles and training centers offering courses. I am also a member of ADAVEC, a french project. This project aims at building a prototype to allow autononous vehicle to adapt continuously its level of autonomy according to the traffic and the security.

Sep2016–Nov2016 **Research Engineer**, *INRIA*, Spirals team.
Development of Nopol, an automatic repair tool in Java, and its plugin in IDEA.

Mar2016–Aug2016 **Internship**, *INRIA*, Spirals team.
Exploring the Perturbability Envelop of Software: study the impact of modifying values during execution on the final output of programs. Potential application are security, optimization or improve the resiliency.

Teaching

June2020–July2020 **Interships supervisor**, *Université Côte d'Azur*.
Supervision of 5 interns working on the social network platform.

March2020–April2020 **Mentor**, *Université Côte d'Azur*.
Product owner of the social network project;
Scrum Master of two teams of 5 developers;

Oct2015–May2018 **Part-time**, *Université Lille 1*.
M1: Building Distributed application;
L3: Oriented object conception;
1^{st} & 2^{nd} year: Support in IT and usage of new technologies.

Misc.

Sept2017–Present **President**, *Université Lille 1*, ADSL.
Organization of social events, promote the PhD.

Feb2017–Sept2017 **Active Member**, *Université Lille 1*, TILDA.
Organization of conferences for Phd students.

IT Skills

Technologies Java/J2EE, Spring, Scala, JavaScript, Angular, Typescript, C/C++, Ruby

Scripts Shell, Python, R

Tools Travis, Jenkins, Git, SQL, LaTeX, Docker, Apache server, Tomcat

Methods Agile, Pull-request based development, Open-source, Continuous integration, Grid5000, DevOps

Publications

Benjamin Danglot, Martin Monperrus, Walter Rudametkin, and Benoit Baudry. An approach and benchmark to detect behavioral changes of commits in continuous integration. *Empirical Software Engineering*, pages 1–37, 2020.

Oscar Luis Vera-Pérez, Benjamin Danglot, Martin Monperrus, and Benoit Baudry. Suggestions on test suite improvements with automatic infection and propagation analysis, 2019.

Benjamin Danglot, Oscar Vera-Perez, Zhongxing Yu, Andy Zaidman, Martin Monperrus, and Benoit Baudry. A snowballing literature study on test amplification. *Journal of Systems and Software*, 157, 2019.

Benjamin Danglot, Oscar Luis Vera-Pérez, Benoit Baudry, and Martin Monperrus. Automatic test improvement with dspot: a study with ten mature open-source projects. *Empirical Software Engineering*, 24(4):2603–2635, 2019.

Oscar Luis Vera-Pérez, Benjamin Danglot, Martin Monperrus, and Benoit Baudry. A comprehensive study of pseudo-tested methods. *Empirical Software Engineering*, 24(3):1195–1225, 2019.

Benjamin Danglot, Philippe Preux, Benoit Baudry, and Martin Monperrus. Correctness attraction: a study of stability of software behavior under runtime perturbation. *Empirical Software Engineering*, 23(4):2086–2119, 2018.

Zhongxing Yu, Matias Martinez, Benjamin Danglot, Thomas Durieux, and Martin Monperrus. Alleviating patch overfitting with automatic test generation: a study of feasibility and effectiveness for the nopol repair system. *Empirical Software Engineering*, 24(1):33–67, 2019.