sCODEV(Smart Collaborative Development Environments) aims at providing a smart collaborative platform for software process development focusing on the following aspects: - Instant developer guidance through the different steps of the process and intelligent monitoring facilities for project managers - Optimum resource assignments on process’s activities i.e., the right skills for the right activity - Scaling-up development teams to reach a global maturity level in general and in the context of a specific project in particular In the context of agile development processes and teams, one challenging aspect is finding the optimum resource allocation. Indeed, usually this is done either based on random allocations, first came first served, whether the developer worked on the task the day before, depending on his skills and expertise, if his code passes the tests successfully (ratio), combining some of these allocation ways together, etc. In order to identify the best strategy for resource allocation, a first and critical step requires defining formally the notion of expertise of developer. Yet another important aspect that we believe to be very important is the notion of developer’s Fluency which represents his ability to complete project’s tasks rapidly and accurately independently of task’s difficulty or importance. Fluency is a broader definition of expertise which may include time spent on the project, quality of the delivered code, and most of all, the global knowledge of the developer on the different modules/components of the application. This global knowledge grows as the developer is assigned with process’s activities that are central to the project. By central activities it’s meant activities that modify/create central and critical components of the application, activities that impact a large number of developers that are related to critical customer’s requirements, etc. All in all, it introduces the notion of activity’s Centrality which in our view worth taking into consideration for an optimum resource allocations but most of all, for achieving the third aspect which is Scaling-up the development team. Indeed, we assume that introducing different components to the developer through process’s activities could raise the global expertise of the team and make it aware of the impact of each modification on the other components. It also makes it easier to make a replacement of a person in case of he leaves the team (keeping in mind the well-known fact that this usually takes longer to integrate a new resource to the team). Ideally, our vision is that a company could simply plug sCODEV to their repositories (code, bug trackers, tests, etc.), inputs the process model with a set of annotations on the tasks (goal, required expertise, complexity, centrality, etc.), same thing on process’s artifacts (importance in the project, link to the global architecture, link to the requirements, etc.) and start the process. The development tool will be then in charge of: - Mining the right developer for the right task and assigning the tasks. - Constantly learning to identify the best resource for a given task by taking into consideration the results of the tests, how the developer’s fluency grows with time and the notion of Centrality. We believe that this is a dynamic data and the result is constantly evolving according to how the project goes. - Depending on the project duration, calculate the best way/resource allocations to scale-up the team. Indeed, some work in the literature demonstrated that it may take 6 to 36 months for developer to scale-up depending on the project’s complexity. So if the project is about to last for 24 months and according to the complexity of process’s activities and the resource in hand (fluency), what would be the best allocation? - And of course, as mentioned earlier, constantly proposing guidance to developers but also to managers for a better monitoring of the process.

Informations complémentaires (Langue 2)
PhD Candidate Profile

The candidate should demonstrate good skills in:

- Writing and doing presentations in English
- Good skills in UML and Java Programming with Eclipse, EMF would be a plus
- Background in Process modeling and execution and knowledge of Agile methods would be a plus

References

- Peter C. Rigby, Yue Cai Zhu, Samuel M. Donadelli, Audris Mockus, Quantifying and Mitigating Turnover-Induced Knowledge Loss: Case Studies of Chrome and a project at Avaya, ICSE 2016, pp (to come)
- Minghui Zhou, Audris Mockus, Developer fluency: achieving true mastery in software projects, SIGSOFT FSE 2010: 137-146
- Filippo Ricca1, Alessandro Marchetto2, and Marco Torchiano3, On the Difficulty of Computing the Truck Factor, Product-Focused Software Process Improvement, Volume 6759 of the series Lecture Notes in Computer Science pp 337-351
- André N. Meyer, Thomas Fritz, Gail C. Murphy, Thomas Zimmermann, Software developers' perceptions of productivity, SIGSOFT FSE 2014: 19-29
- Matthieu Foucault, Marc Palyart, Xavier Blanc, Gail C. Murphy, Jean-Rémy Falleri, Impact of developer turnover on quality in open-source software, ESEC/SIGSOFT FSE 2015: 829-841
- Cédric Teyton, Marc Palyart, Jean-Rémy Falleri, Floréal Morandat, Xavier Blanc, Automatic extraction of developer expertise, EASE 2014: 8:1-8:10