

Requirements Engineering in the DevOps Era

Sébastien Mosser
Université du Québec à Montréal
Montréal, Canada
mosser.sebastien@uqam.ca

Jean-Michel Briel
University of Toulouse, IRIT
Toulouse, France
bruel@irit.fr

Abstract—Traditional requirements engineering does not benefit from modern software development techniques. The goal of this tutorial is to revisit requirements elicitation and to bridge the gap between traditional requirements engineering and modern software development. This tutorial will provide the attendees with concrete and immediately applicable guidance by demonstrating how to operationalize a fully-fledged toolchain going from user stories to automated acceptance testing using open-source tools.

I. MOTIVATIONS AND OBJECTIVES

We propose a tutorial on Requirements Engineering in an DevOps context that aims at exploring the boundaries between requirements, specifications, stories, scenarios and tests. It follows the recent work started in the RE community about agility, from a practical point of view [2], [5].

Revisiting requirements elicitation and bridging the gap between traditional requirements engineering and modern software development (highly based on continuous integration and tests), this tutorial will demonstrate how to operationalize a fully-fledged tool chain going from user stories to automated acceptance testing using open-source tools. This is applicable to industrial practitioners as we will rely on state of the art tools, and link agile requirements to formal requirement engineering methods.

We will capitalize on the well known a concept of user stories and epics and their usage for expressing users requirements. The objective of the tutorial is to focus on the automation of the evaluation of such requirements based on definition of done acceptance criteria. We will demonstrate how such requirements can be tracked in a project management tool, and linked to source code development. At the source code level, we will demonstrate how the stories and the associated acceptance scenarios can be modeled using the Gherkin¹ language [7], and linked to classical unit tests to automate their validation. Finally, a continuous integration environment will be deployed using Docker to link together the different tools and offer an automated pipeline for software developers, bridging the gap between requirements and code development.

II. FORMAT AND SERVICES

In this half-day tutorial, we will focus on the following topics:

- Quick reminder of requirements in an DevOps context: how one can integrate requirement analysis in a DevOps loop?
- Using User Stories as a main requirement artifact [1] and bridging the gap between them and the code through automation based on proven methods and techniques such as BDD
- Criticality of acceptance criteria definition and validation using Cucumber [7] and JUnit [3]: How to support and automate stories validation? How to leverage unit testing (intended to developers) and support QA tasks?
- Role of continuous integration [6] in the development process to bind requirements to operational context: how to quickly deploy a piece of software, and maintain traceability of the stories in the source code and the associated features?

We propose to follow a hands-on approach, where participants will experiment the different concepts directly using prepared material. Each topic will be exposed using the following (and systematic) approach: first, a quick description of the task to be performed by the participant, then a practical exercise to experiment the proposed approach and finally a retrospective used to link the outcome of the exercises with formal requirements engineering methods.

The audience is expected to work on the exercises using their own laptops, where Git and Maven must be installed. We will provide a link to a public repository containing materials, exercises and setup instructions. An Internet connection will be mandatory during the workshop to access to the materials.

III. TARGET AUDIENCE

The target audience are software engineers, modelers, requirements engineers, or teachers. The first key idea of the tutorial is to allow curious RE practitioners to experiment practically how software developers interact with requirements when working in an agile context. We will not focus on any particular agile method, and instead experiment the role of requirements engineering with respect to the pillars of agility, i.e., delivering value to customers in a continuous and simple way. The second key idea is to reconcile software developers with requirements engineering, by concretely illustrating how useful they can be for the development of quality software. Most of the time, software developers have very little consideration or even knowledge of the requirements. We will demonstrate that they are linked with user stories (in the

¹See <https://github.com/cucumber/cucumber/wiki/Gherkin>.

agile sense) and tests. The tutorial will also demonstrate how to successfully exploit these artifacts to reify such a link at the developer level. As this tutorial is designed for the RE community, we will focus on the coverage of the DevOps loop starting at this level, and going up to the deployment in an operational context.

IV. TUTORIAL HISTORY

As far as we know, such tutorial has only been proposed at RE'18 [4], but the focus was there on Agility and expressing requirements as User Stories. The authors have since experimented and improved this approach and they have long experience on the concepts that will be introduced and experimented by the attendees. They have also a good experience in workshops collaborations and animation. This tutorial proposal is built on top of a seven years experience in teaching agile software development at both graduate and undergraduate level at UQAM, University Code d'Azur, or University of Toulouse, in interaction with professional practitioners. To bind this experience to the requirement engineering research community, we will leverage Jean-Michel Bruel's experience in teaching SysML and requirements engineering in different Masters. His research topics include requirements formalization and traceability. The tutorial will only address pragmatic and useful features in that matter.

V. PRESENTER'S BIO

Sébastien is Professor of Software Engineering at *Université du Québec à Montréal* (UQAM). His research interests cover DevOp, software development and separation of concerns mechanisms, with a particular focus on domain-specific languages, model-driven engineering and software composition. Since 2007, he has contributed to the software composition field by providing composition mechanisms used to tame variability in several domain, such as cloud, internet of things, business processes and microservices. Sébastien was Virtual Chair of MODELS 2020, and has been part of the organization and program committees of several conferences and workshops related to variability management (e.g., VaMOS, SPLC), model-driven engineering (e.g., MODELS, MODELWARD) and software development (e.g., ICSE artefacts evaluation), including industrial events.

Jean-Michel Bruel is Professor at the University of Toulouse (France). He is leading the SM@RT team, specialized in models and language engineering. He has animated several tutorials at MODELS or RE. He has been teaching Agile Methods, Requirements Engineering, and Model-Based Systems Engineering for more than 12 years.

VI. PUBLICITY

Both authors have been publicity chairs of major conferences such as MODELS, RE or Modularity. They are also involved in requirements and software engineering communities. Hence they will intensively promote the tutorial through their networks (e.g., mailing lists, twitter).

REFERENCES

- [1] Mike Cohn. *User Stories Applied: For Agile Software Development*. Addison Wesley Longman Publishing Co., Inc., USA, 2004.
- [2] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa. Requirements engineering challenges in large-scale agile system development. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 352–361, 2017.
- [3] Jeff Langr, Andy Hunt, and Dave Thomas. *Pragmatic Unit Testing in Java 8 with JUnit*. Pragmatic Bookshelf, 1st edition, 2015.
- [4] Guenther Ruhe, Walid Maalej, and Daniel Amyot, editors. *26th IEEE International Requirements Engineering Conference, RE 2018, Banff, AB, Canada, August 20-24, 2018*. IEEE Computer Society, 2018.
- [5] S. Saito, Y. Iimura, A. K. Massey, and A. I. Antón. How much undocumented knowledge is there in agile software development?: Case study on industrial project using issue tracking system and version control system. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 194–203, 2017.
- [6] S. Stolberg. Enabling agile testing through continuous integration. In *2009 Agile Conference*, pages 369–374, 2009.
- [7] Matt Wynne and Aslak Hellesoy. *The Cucumber Book: Behaviour-Driven Development for Testers and Developers*. Pragmatic Bookshelf, 2012.