Reconciling Requirements and Continuous Integration in an Agile Context

Sébastien Mosser, Jean-Michel Bruel

Reconciling Requirements and Continuous Integration in an Agile Context

Sébastien Mosser
Université Côte d'Azur, CNRS, I3S, France
Email: mosser@i3s.unice.fr

Jean-Michel Bruel
University of Toulouse, IRIT/CNRS, France
Email: bruel@irit.fr

I. Motivations & Objectives

The RE'18 theme is: "*Crossing Boundaries and Increasing Impact*". In this context we propose a tutorial on Requirements in an Agile 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 [1], [2].

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 first focus on the notion of user stories and epics to express users requirements, and how to evaluate such requirements based on definition of ready and definition of done acceptance criteria. We will then 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 Gherkin[1] language [3], 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:

- Role of requirements in an agile context: how does the requirements elicitation phase interfere with agile development processes? Is being agile a way to bypass requirement analysis, or a way to integrate it in a daily routine? What is the role of a *product owner* with respect to a requirement engineer?
- Writing User stories as a main requirement artifact [4]: what is a user story, and how does it fit in the development process? What are the different levels of details for a story? How to link a story to a formal requirement artifact? When can a story be considered as ready for development?
- Criticality of acceptance criteria definition and validation using Cucumber [3] and JUnit [5]: 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 agile sense) and tests. The tutorial will also demonstrate how to successfully

[1] See https://github.com/cucumber/cucumber/wiki/Gherkin.

exploit these artifacts to reify such a link at the developer level.

IV. Tutorial History

As far as we know, such tutorial has never been proposed in a software engineering conference. The authors have never experimented this particular format but 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 five years experience in teaching agile software development at both graduate and undergraduate level at Université Côte d'Azur and 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. Presenters's Bio

Jean-Michel Bruel is Full 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 and Model-Based Systems Engineering for more than 10 years. One of its research interest is to define the semantics of traceability links.

Sébastien Mosser is Associate Professor at Université Côte d'Azur (France). He is coordinating the Software Architecture specialization of the Computer Science MSc, and teaches software engineering courses at both graduate and undergraduate levels. His research interests cover software development and separation of concerns mechanisms.

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] R. Kasauli, G. Liebel, E. Knauss, S. Gopakumar, and B. Kanagwa, "Requirements engineering challenges in large-scale agile system development," in *25th IEEE International Requirements Engineering Conference, RE 2017, Lisbon, Portugal, September 4-8, 2017*. IEEE Computer Society, 2017, pp. 352–361. [Online]. Available: https://doi.org/10.1109/RE.2017.60

[2] 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 *25th IEEE International Requirements Engineering Conference, RE 2017, Lisbon, Portugal, September 4-8, 2017*. IEEE Computer Society, 2017, pp. 194–203. [Online]. Available: https://doi.org/10.1109/RE.2017.33

[3] M. Wynne and A. Hellesoy, *The Cucumber Book: Behaviour-Driven Development for Testers and Developers*. Pragmatic Bookshelf, 2012.

[4] M. Cohn, *User Stories Applied: For Agile Software Development*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[5] J. Langr, A. Hunt, and D. Thomas, *Pragmatic Unit Testing in Java 8 with JUnit*, 2nd ed. Dallas, TX: Pragmatic Bookshelf, 2015. [Online]. Available: https://www.safaribooksonline.com/library/view/pragmatic-unit-testing/9781680500769/

[6] S. Stolberg, "Enabling agile testing through continuous integration," in *2009 Agile Conference*, Aug 2009, pp. 369–374.

[7] *25th IEEE International Requirements Engineering Conference, RE 2017, Lisbon, Portugal, September 4-8, 2017*. IEEE Computer Society, 2017. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8048783