Teaching the Essence of Software Development

Teaching the Essence of software development

Paolo Ciancarini
University of Bologna and Innopolis University

Marcello Missiroli
University of Bologna

***Abstract*—We present our plan to introduce Essence in our courses in software engineering, including a plan to evaluate our results and develop Essentialized tools.**

***Index Terms*—Software Engineering Education, Serious Games, Software Engineering Tools, Essence language.**

I. Introduction

Essence is a new visual framework including a kernel and a language suitable for describing software development methods and processes [1]. It is comparable to UML, in that it aims at providing a common and versatile means of communicating the structure and the evolution of complex entities.

UML had a strong impact on the teaching of software development, becoming a standard visual formalism used in several software engineering classes; it also inspired model-driven software engineering. In addition, it is also used in CS education, to teach, clarify and educate developers-to-be. Like UML, Essence is also an OMG standard, and this should facilitate its diffusion.

We argue that Essence, like UML, has the potential to be useful as a teaching tool as much as an industrial one. However, one cannot blindly introduce Essence in a CS course; it is a powerful framework that we have to analyze and introduce in the teaching routine having in mind specific goals and procedures, lest it become an empty tech-oriented show-off.

In this paper we describe a research path that explores the adoption of Essence to teach SE concepts to both undergraduate and graduate students in Software Engineering. This path comprises the following main objectives:

1) Exploiting Essence as a teaching guide for the syllabus of a course in software development; we plan to use the Essence book [1] as a basis for the course.
2) Developing some Essence-enabled tools, eg. for project management or software repository.
3) Using an Essence-based method and some Essence-enabled tools for students' projects.
4) Evaluating our experience using Goal-Question-Metrics (GQM) and the Technology Acceptance Model (TAM) developed by Davis [2]. TAM is a well-known theory intended to explain technology adoption on a general level in the domain of information systems; it links technology acceptance two two parameter: Perceived Usefulness (PU) and Perceived Ease of Use (PEOU) and their correlation.

The structure of this paper is the following: Section II presents our proposal on using Essence as a teaching tool; Section III describes what is intended by Essence-enabled tool and how we plan to develop one using open source components; Section IV explains how we will use Essence in students' projects; Section V describes how we will evaluate results.

II. Essence as a teaching tool

Software Engineering is a discipline difficult to teach, and many scientific conferences are devoted to this topic.

In the last few years we have been engaged in teaching software development using an agile approach [3]. We, like others, have exploited an approach including serious games for training students in agile practices. Serious games are popular in several contexts, including corporate business, education, healthcare, and tourism management. Starting from the observation that most software development activities can be simulated in class or "played", serious games for Agile training and coaching are becoming very popular to engage software developers [4].

Essence is a visual language for defining methods and practices common to all software engineering. It potentially provides students, developers and scholars with a common ground, irrespective of the actual methodology used. Essence is therefore very interesting from an educational point of view, as it offers students an attractive method to clarify, visualize and memorize the complex concepts used in SE. The Essence book itself [1] is organized with learners in mind; it uses a simple language, provides several practical examples, and the topics are logically ordered. Still, in our view, the inclusion of the Essence language - as any technology - cannot be done overnight, but need to be integrated within the existing course materials, adapted to the teacher's peculiarities and possibly expanded. Ultimately, its usefulness should be tested and validated in some way, for instance using essentialized serious games.

The SE course we teach has as prerequisites the ability to program in an OO language. For the next edition we plan to introduce the following changes, beginning on the semester during Fall 2020.

- Introduce the Essence language during the first part of the course, devoted to the software process, and adapt both teaching material and course structure accordingly;
- survey a few development methods - eg. waterfall, Scrum, and more - using the Essence language. The Essence book only describes Scrum (to be precise, a "Scrum

Lite" method) and some related practices providing a strong Agile-oriented mark on the course. However, when teaching it is important describe several other methods, such as the classic Waterfall model, at the very least for historical reasons. We plan to describe them using the Essence language;
- as we have developed the core of an open source Project Management System - [5], we intend to "essentialize" both the methodology and some services available within the system itself.
- develop essentialized serious games based on Agile practices; we have some experience with these games [6], we will ask students to help us.

III. Enabling Essence in existing tools

In modern software development, using efficient tools and integrated environments is a mandatory practice. Since Essence is a young proposal, not many tools are currently available; the site Sematacc https://sematacc.herokuapp.com/ includes the following ones:

1) **Semat Practice Library** it provides all basic and advanced Essence elements organized as an integrated hypertext. Useful to clarify the large definitions and elements of the language, allowing download of printable cards.
2) **Semat Accelerator** [7]. This web-based project it relatively old, but offers a nice visual interface to some Essence entities, particularly those linked to project status.
3) **Essencery** [8]. This allows the user to create new Essence methods by combining and labeling Essence language symbols.

In our view, there are several practical issues tied to these tools:

- The tool is not linked to actual project development artifacts (such as source code or project managements analytics) - all updates must be done manually (this is true for all)
- The tool does not use the visual language of Essence (1,2)
- The tool is complicated (1,3)
- The tool has a limited scope and is difficult to extend (1,2)
- The tool is not open source (3)

There are also other tools, like [9], [10]. The former presents a simple infrastructure for project monitoring and steering using Essence. The latter presents the Essence Navigator, a simple execution environment for Essence kernels as defined by the operational semantics of the Essence specification.

We are planning to develop a tool with the following goals in mind

1) Licensed as Open Source software
2) Easy to integrate with some existing software management tools, especially open source ones like Taiga and Gitlab (via plugins or direct integration) and our own CAS [5].
3) Extensive use of Essence visual language in combination with an agile method inspired to Scrum.

IV. Essentializing students projects

As part of the course, we intend to promote the use of Essence throughout the course among students. This will start by using Essence in preparatory (toy) projects, serious games and workshops. Essence will be a required part of the capstone project, as we will ask to describe the status of the project and the methodology in use at different points in time.

V. Evaluation

All these changes and novelties need to be properly evaluated. For tools, we will measure:

- Adherence to Essence standards.
- Integrability with the existing tool(s).
- Usability, using ad-hoc usability tests.

For changes in the teaching syllabus and methodology, we will use a variety of methods and metrics, such as:

- Specific evaluation of the Essence language using a TAM [2] evaluation gathering technique;
- Holistic evaluation of changes via GQM methodology;
- Comparison of institutional students' feedback of previous years with the current year.

Acknowledgements. We thank for support the Consorzio Interuniversitario per l'Informatica (CINI) under the AMINSEP contract.

References

[1] I. Jacobson, H. Lawson, P. Ng, P. McMahon, and M. Goedicke, *The Essentials of Modern Software Engineering*, ser. ACM Books. Morgan & Claypool Publishers, 2019.

[2] F. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.

[3] M. Missiroli, D. Russo, and P. Ciancarini, "Learning Agile software development in high school: an investigation," in *Proc. 38th International Conference on Software Engineering*. Austin, Texas: ACM, 2016, pp. 293–302.

[4] A. Przybyłek and M. Zakrzewski, "Adopting collaborative games into agile software development," in *Proc. 13th Int. Conf. on Evaluation of Novel Approaches to Software Engineering (ENASE 2018)*, ser. Communications in Computer and Information Science, vol. 1023. Springer, 2018, pp. 119–136.

[5] P. Ciancarini, M. Missiroli, F. Poggi, and D. Russo, "An open source environment for an agile development model," in *IFIP International Conference on Open Source Systems*, ser. Advances in Information and Communication Technology, vol. 582. Springer, 2020, pp. 148–162.

[6] P. Ciancarini and M. Missiroli, "Educating students to team-based software development using agile games," (in preparation), 2020.

[7] D. Graziotin and P. Abrahamsson, "A Web-based modeling tool for the SEMAT Essence theory of software engineering," *arXiv preprint arXiv:1307.2075*, 2013.

[8] A. Evensen, K.-K. Kemell, X. Wang, J. Risku, and P. Abrahamsson, "Essencery-a tool for essentializing software engineering practices," *arXiv preprint arXiv:1808.02723*, 2018.

[9] C. Péraire and T. Sedano, "State-Based Monitoring and Goal-Driven Project Steering: Field Study of the SEMAT Essence Framework," in *Proc. Int. Conf. on Software Engineering (ICSE'14)*, 2014.

[10] J. Pieper, O. Lueth, M. Goedicke, and P. Forbrig, "A case study of software engineering methods education supported by digital game-based learning: Applying the SEMAT Essence kernel in games and course projects," in *Proc. IEEE Global Engineering Education Conference (EDUCON)*, 2017, p. 1689–99.