Quality Requirements in Agile as a Knowledge Management Problem: Balancing Just-In-Time and In-Good-Time RE

Eric Knauss, others?
Computer Science and Engineering
Chalmers | University of Gothenburg
Gothenburg, Sweden
{authors}@chalmers.se, {eric.knauss}@cse.gu.se

Kurt Schneider, others?
Software Engineering Group
Leibniz Universität Hannover
Hannover, Germany
ks@inf.uni-hannover.de

***Abstract*— Eric ►*very drafty, to set the scene*◄ Managing non-functional requirements in agile projects may look all-JIT at first glance. However, some non-functional requirements cannot be raised and met on the spot. We argue that effective JIT-RE in some cases requires building on a solid foundation of other quality requirements that are not invented and changed just in time. Different roles and different stakeholders are responsible, and the tasks at hand determines how they can be handled.**

Thus, it is not sufficient to focus on JIT requirements engineering alone. If an Learning Software Organization (LSO) intends to increase agility and speed up system development, it needs a holistic concept for managing this knowledge. The assumed co-existence of JIT and solid non-functional requirements needs a structure for eliciting, storing, and managing quality requirements. If JIT is supposed to work, it needs to be complemented by non-JIT requirements, and by an infrastructure to accommodate both.

We claim it is difficult to handle JIT requirements, since quality requirements often have a long-term perspective.

We propose a knowledge-management framework that facilitates JIT-RE by providing knowledge management structures and counterparts to represent – and update – the knowledge aspect of non-functional requirements. Kurt ►*Which properties of the framework are important so that it supports JIT and sustainable evolution?*◄

***Index Terms*—just-in-time RE, quality requirements, managing requirements knowledge**

I. Introduction

Eric ►*4pages!*◄

Eric ►*title, abstract, intro to finish on page 1*◄

Eric ►*Abstract, Intro: Basic assumption JIT+long-term; this is a KM problem*◄

- NEW: There exist different schools of knowledge management and we can conclude that there are alternatives to a technocratical, database centric approach. Behavioral and business perspectives are important in agile. (argue with EKSE book, p60)
- NEW: Business value in agile methods does very well fit into this view.

  Eric ►*Trying to write some chain of argument*◄

  a) Context: More and more system development companies are turning towards agile...

  b) Quality requirements have JIT and long-term aspects: and we cannot support one without knowing about the other. A holistic knowledge management approach is needed.

  c) Agile System Development needs a user value and a system understanding perspective: This is based on [**?**].

II. Background and Related Work

Eric ►*Classic KM, RBS, Cockburn...*◄

- RBS's QUPER model relates quality requirement to competitors: Only good enough to be leader/follower. Implies long-term perspective.
- Nonaka: Externaliziation, socialization (combination (=tracing), internalization (=understanding system)
- Usabilty/security/safety: qualities differ, also their JIT behavior.
- Cockburn: two games. Be efficient know, but also in a good position for the next release/project
- Doors as a requirements database, a knowledge base of an LSO

III. Towards a KM Framework

Constraints for such a framework, add two examples as boxes.

- Eric's 4 field model (portfolio: user value - system knowledge / JIT - Long-term)
- Show two examples, Security (Kurt), Safety (MARK-C WS on regulations)
- SecVolution: Keep evolving system secure

IV. Conclusion and Outlook

Some research roadmap perhaps?

Acknowledgments

We thank all participants in this study for their great support, deep discussions, and clarifications. This work was supported by Software Center Project 27 on RE for Large-Scale Agile System Dev. and the SIDA BRIGHT project.

References