Requirements Engineering: Leading Challenged Customers to Success in Large-Scale Agile Development

Elke Hochmüller

Carinthia University of Applied Sciences, Primoschgasse 8,
9020 Klagenfurt, Austria
E.Hochmueller@cuas.at

Extended Abstract

Agile development methods emerged as a light-weight alternative to strictly phase-driven processes. These methods aim at short time-to-market and are based on close customer cooperation, small and mostly self-organizing teams, minimal bureaucracy, minimal documentation effort, and incremental delivery with short iteration cycles and several small releases.

In contrast to phase-driven processes with the tedious, up-front phase of acquiring, analyzing, negotiating, writing and agreeing on a complete requirements specification document, agile projects rely on continuous face-to-face communication with the customers to determine the "just-in-time" requirements for each iteration cycle. This allows for the more realistic case that not all requirements have to be readily stated and specified from the beginning of the project, but can evolve and manifest over time.

Success stories showed that agile processes and practices are the perfect alternative to inflexible phase-driven processes in case of small-sized projects. Large-scale software development, however, poses still a challenge to agile teams and customers. Some of the challenges customers of large-scale agile projects are faced with are:

- **Mismatch in planning horizons.** Customers usually have to plan their budget in the long run on basis of the full system to get a qualified authorization for project kick-off. Agile methods, however, do not foresee long-term planning and cost-estimation. The mismatch in planning horizons progressively increases with the size of the project. Hence, the risk of a possibly incomplete system after exceeding the authorized budget is totally up to the customers of agile projects. [2]
- **Burden of responsibility.** Particularly in multi-stakeholder projects, the preferably on-site customer representative can soon be overburdened in his/her role as omnipresent as well as omniscient person who not only has to acquire, understand, and describe the essential client requirements but also has to deal with conflicting requirements, diverse expectations, and (un)consciously unstated knowledge (tacit knowledge) of his/her colleagues. [2]
- **Late quality requirements.** Agile processes explicitly welcome late or even changing requirements. This is certainly beneficial in case of functional requirements. However, certain types of quality requirements (e.g. security, performance, availability, reliability, ...) will substantially influence decisions on the architecture of a system [1, 7]. The related actual requirements have to be

elicited as early as possible in order to avoid misdirected investments due to wrong architectural decisions. However, quality requirements will usually not be the first ones to be required by agile customers who will primarily come up with functional requirements that are obvious and easier to be acquired.

- **Continuous migration activities.** Due to short development cycles, new or altered system components will be delivered to be integrated into a running system within short periods of time. Continuous data migration and user training will be some of the consequences. In the meanwhile, undisturbed operation has to be guaranteed. [2]
- **Development versus maintenance effort.** In contrast to phase-oriented software processes, agile processes usually result in a rather early delivered first version of a system with continuous changes after installation. Hence, the main effort is shifted from the development phase towards a long-lasting maintenance phase in the environment of the customer with all the already mentioned high-frequency migration activities. [2]

What can we do to support the customer in order to overcome the above and other challenges and to enable the customer to be a valuable partner for agile development teams?

What about reusing our knowledge and competence in requirements engineering to bridge the gap between customers and agile developers by means of requirements engineering practices to be adapted to large-scale agile development projects?

Related activities to be performed by a requirements engineer acting on behalf of the customer in agile development projects can be: [3]

- ***Rough long-term cost estimation.*** Particularly in large-scale projects, a preliminary time-schedule and gross budget prediction based on an early analysis of the system's purpose and its main use cases will be inevitable to enable clients to even authorize an agile software development process.
- ***Identification of relevant stakeholders.*** In multi-stakeholder projects, a skilled requirements engineer will be able to help the client in selecting the adequate representatives of all the different viewpoints which have to be considered during the project.
- ***Timely acquisition, analysis, documentation, and validation of quality requirements.*** An experienced requirements engineer will be able to focus on the "right" requirements at the "right" time (in terms of both requirements quality as well as quality requirements) assuring the proper treatment of essential key requirements determining the decision on an adequate software architecture.
- ***Iterative identification of functional requirements.*** A requirements engineering professional has to provide customers with continuous support in eliciting, negotiating, prioritizing, and precisely expressing the subset of functional requirements to be implemented during each development cycle.
- ***Definition of "security walls".*** According to traditional principles of incremental delivery [4], the functional requirements shall be packaged in such a way that possible interference between the deliverables of different development cycles will be avoided in order to minimize migration effort when installing the next code increment.

- ***Test case development and test support.*** Traditional requirements engineering activities already include the definition of test cases. In large-scale agile development projects, a requirements engineer has to provide the customer with additional support in the areas of integration and acceptance tests of the increments delivered.
- ***Support in integrating new releases.*** Another "new" task for a requirements engineer in agile projects will be the assistance in rolling out the incrementally delivered components to the productive system at the site of the customer.

The kind of fulfilling the above activities will vary in detail depending on the actual agile method to be applied. Furthermore, the extent of a requirements engineer's participation in an agile project can vary according to clients' needs; be it as a consultant on occasion, or as a customer coach [6], or as part of a customer pair [7] together with a customer representative, or as "the" customer representative.

However, it is worthwhile to further investigate in requirements engineering as a facilitator of large-scale agile development projects.

References

1. Albin, S.T.: The Art of Software Architecture: Design Methods and Techniques. Wiley. Indianapolis (2003)
2. Hochmüller, E., Mittermeir, R.T.: Agile Process Myths, Proc. APSO'08: Scrutinizing Agile Pracitces, or "Shoot out at Process Corral", Leipzig, pp. 5–8. (2011)
3. Hochmüller, E.: The Requirements Engineer as a Liaison Officer in Agile Software Development. In: 1st Agile Requirements Engineering Workshop. ACM, Lancaster (2011)
4. Hsia, P., Hsu, C., Kung, D.C., Yaung, A.T.: The Impact of Incremental Delivery on Maintenance Effort: An Analytical Study, Proc. ESEC'95: 5th European Software Engineering Conference, Sitges, pp. 405–422 (1995)
5. Martin, A., Biddle, R., Noble, J.: XP Customer Practices: A Grounded Theory. Proc. AGILE'09: Agile Conference, Chicago, pp. 33–40 (2009)
6. Martin, A., Biddle, R., Noble, J.: XP Customer Team: A Grounded Theory. Proc. AGILE'09: Agile Conference, Chicago, pp. 57–64 (2009)
7. Orr, K.: Agile Requirements: Opportunity or Oxymoron? IEEE Software. 21(3), pp. 71–73 (2004)