What Language Does Agile Speak?

Rashina Hoda, James Noble, and Stuart Marshall

School of Engineering and Computer Science,
Victoria University of Wellington,
New Zealand
{rashina,kjx,stuart}@ecs.vuw.ac.nz
http://www.ecs.vuw.ac.nz

Abstract. Collaboration-intensive Agile practices are dependent on the development team understanding the customer's perspective and requirements. Through a Grounded Theory study of Agile teams in New Zealand and India, we discovered that a gap between the teams' *technical* language and the customers' *business* language poses a threat to effective team-customer collaboration. We describe this language gap and the 'Translator' role that emerges to bridge it.

Keywords: Agile Software Development, Customer Collaboration, Language Gap, Translator, Grounded Theory.

1 Introduction

Agile software development requires regular collaboration between development teams and their customers, in an effort to build software solutions that meet the customers' needs [1,4,5]. Collaboration-intensive Agile practices, such as release and iteration planning, user acceptance testing, and reviews; are unlikely to succeed unless development teams are able to effectively understand the customers' prespectives and requirements [3,7]. Through a Grounded Theory [2] study of 20 Agile practitioners across 12 different software organizations in New Zealand and India, we discovered this gap between the teams' *technical* language and the customers' *business* language threatens effective team-customer collaboration and found an informal *Translator* role that emerges to bridge this language gap.

2 The Language Gap

As a result our data analysis, we discovered that development teams and their customers use different languages when collaborating on Agile projects. While the development teams use a more *technical* language composed of technical terminology, their customers are used to a more *business* langauge composed of terminology from the customers' business domains. The language gap between development teams and their customers poses a threat to effective team-customer collaboration by limiting their understanding of each other's perspectives. Translation between the two languages was found to be necessary in order to maximize the effectiveness of collaboration-intensive Agile practices and led to the emergence of an informal *Translator* role.

A. Sillitti et al. (Eds.): XP 2010, LNBIP 48, pp. 387–388, 2010.

3 Translators — Bridging the Language Gap

The *Translator* is a development team member responsible for facilitating communication and collaboration between the team and their customers by translating between their respective languages. Initial analysis suggested that the role of the *Translator* was best suited to Business Analysts because of their perceived ability to act as a bridge between the two languages [4]. As our research progressed, we found that most members of mature Agile teams (fluent in use of Agile practices, for usually more than a year) were bilingual: all able to play the *Translator* role. Several participants believed that certain tools and techniques help in becoming a successful *Translator*:

Using a Dictionary. The *dictionary* was an online editable document (wiki) populated by the customers with business terms, their meaning, and their context of use. These business terms were translated directly into code by the team using the same variable names, providing one-to-one mapping between the customers' *business* terms and their *technical* implementation for a given project.

Using Story Cards. Story cards were used by customers to describe requirements in *business* language. The team can be assisted in translating these story cards into *technical* tasks by (a) assigning a coach to guide new teams on the use of story cards and (b) frequent release of software to ascertain the correctness of the translation through customer feedback.

Using Iterative Reasoning. Another key to becoming a *Translator* was found to be the power of questionning proposed technical solutions repeatedly till the abstract business reasoning behind the technical details was unraveled and were clearly aligned with their business drivers. "*Why do we need that database back up procedure? ...right down at the technical level [asking] the question why, why, why till...you'll eventually discover there's a good business reason for having it.*" (Senior Agile Coach, NZ)

Promoting Cross-functionality. Interactions between members from diverse disciplines fosters understanding of the project from multiple perspectives [6]. As the team learns to understand their customer's perspective, they achieve greater levels of cross-functionality and are able to translate between their respective languages.

References

1. Dybå, T., Dingsoyr, T.: Empirical Studies of Agile Software Development: A Systematic Review. J. Inf. Softw. Technol. 50(9-10), 833–859 (2008)
2. Glaser, B., Strauss, A.L.: The Discovery of Grounded Theory. Aldine, Chicago (1967)
3. Grisham, P.S., Perry, D.E.: Customer relationships and Extreme Programming. In: Proceedings of HSSE 2005. ACM, New York (2005)
4. Hoda, R., Noble, J., Marshall, S.: Organizing Self-Organizing Teams. In: Proceedings of ICSE 2010, Cape Town (to appear May 2010)
5. Highsmith, J., Fowler, M.: The Agile Manifesto. Soft. Dev. Magazine 9(8), 29–30 (2001)
6. Takeuchi, H., et al.: The new new product development game. Harvard Bus. Review (1986)
7. Misra, S.C., Kumar, V., Kumar, U.: Identifying some important success factors in adopting agile software development practices. J. Syst. Softw. 82(11), 1869–1890 (2009)