Empirical Evidence Principle and Joint Engagement Practice to Introduce XP

Lech Madeyski[1], Wojciech Biela [2]

[1]Institute of Applied Informatics, Wroclaw University of Technology, POLAND
Lech.Madeyski@pwr.wroc.pl, http://madeyski.e-informatyka.pl
[2]ExOrigo Sp. z o.o., Krucza 50, 00025 Warsaw, POLAND
Wojciech.Biela@exorigo.pl, http://www.biela.pl

Abstract. Bringing software process change to an organisation is a real challenge. The authors have shown a sample attempt to carry out a process change and then reflected on its results and context. The present reflection points to a need for a set of principles and practices that would support the fragile process of introducing agility. For a start, the authors propose the Empirical Evidence principle exemplified using DICE® and the practice of Joint Engagement of the management and the developers. Both are results of a real-world process change case study in Poland.

1 Background

The company under study is medium-sized (below 200 employees) and employs 30+ programmers in various cities in Poland.

This paper focuses on one project developed in a 2-year period by a remote team. The project was developed by 3 programmers (the team's total was 8) using the Java technology stack. It was a web application, a B2C platform for a trust fund agent.

The project involved various problems, e.g. outdated requirements, system architecture structured but fragile, etc. The team was not using common best practices like loosely coupled code. Usually there was chaotic "code and fix" cowboy coding.

The problems were addressed using a toolkit of agile techniques. One of the authors joined the team to seek out and address problems. At that time, though he had limited experience, he had a strong belief based on of-the-books knowledge and academic projects (e.g. e Informatyka [1]) directed by the co-author of this paper.

Test-Driven Development (TDD) was new and the developers engaged in the project found it to be very effective and rewarding. **Refactoring** was used in the past, but not explicitly and often. Bringing it to a new level let the team implement radical changes without endangering the project. **Pair Programming** (PP) results were inconclusive and managers were reluctant to it. Nevertheless, it helped to share the team's knowledge and halved the time of bringing a new person to the project. **In-process design** sessions required a lot of coaching. **Problem Decomposition** was the most successful technique brought to the team. Dividing the problems into pieces and solving them at that level was really refreshing. **Continuous Integration** and task

automation was an obvious benefit. **Darts** or similar group toys are a must-have for any development team. It was another "motivation for regular breaks"[2] and glued the team together. **Communication** still is an issue because the team is remote. Nevertheless, wiki and encouraging people to use Skype helped to some extent.

2 Rolling the DICE® Twice

The authors used the DICE framework [3], created by The Boston Consulting Group, to confirm that the adoption of agile practices actually increases the project's chance of success. The same metric is now used to convince the organisation's top management to conduct further changes.

The DICE framework is a simple empirical evidence-based formula for calculating how well an organisation is or will be implementing its change initiatives [3]. The DICE® framework comprises a set of simple questions that help score projects on each of the four factors: project duration (D), team's integrity (I), commitment of managers (C1) as well as the team (C2), additional effort (E) required by the change process. In DICE®, a project with an overall score between 7 and 14 is considered a *Win*, between 14 and 17 is a *Worry* and between 17 and 28 is a *Woe*. The DICE® formula is $D + 2 * I + 2 * C1 + C2 + E$.

Duration (D) *Before the change (score 2):* There was no notion of iterations, nor project rhythm. *After the change (score 1):* 1–2-week iterations with reviews during the Planning Game sessions. **Integrity (I)** *Before (score 3):* The previous project team leader was not a team leader, he was a sound programmer, but lacked social skills and the will to innovate. *After (score 2):* The current project team leader is capable and eager to implement new ideas. The team is quite self-organising. **Management Commitment (C1)** *Before (score 3):* Management did not involve enough resources for the changes, mainly because they felt the change should take less than the programmers had said. *After (score 2):* Changes took less time (there were fewer bugs), so the management was more supportive. **Team Commitment (C2)** *Before (score 3):* Changes to the project usually met with resistance, because changes usually meant that something else would break then. *After (score 1):* Changes met with much less resistance due to the ability to refactor in a controlled environment (test case safety net). **Effort (E)** *Before (score 3):* Effort required by the change was normal. Upfront design, followed by coding, testing and bug-fixing cycles. *After (score 2):* The overall effort was not as great as before, because although unit-testing and pairing were applied, there was no long design phase and there was less bugfixing. The DICE® score *before* (20) and *after* the change (12) moves the project from the *Woe zone* into the *Win zone*. This suggests that the introduction of agile practices resulted in an environment which facilitates changes more adequately.

The DICE® framework may be also used to measure the responsiveness to change in other contexts, as outlined in [4]. Making use of the experiences of Ziółkowski and Drake [5], the authors used DICE® again – this time to measure the above organisation's capability for carrying out process changes.

Duration *(score 1).* There was no notion of a formal review as it did not fit the agile environment the author was trying to create. Reviews were done frequently by

him and his superiors in a more casual fashion. **Integrity** *(score 3)*. The change leader (the author) was not given enough resources to achieve his goal. The author was then seriously lacking practical knowledge on how to implement development agility in a concrete situation. **Management Commitment** *(score 3)*. There was some change support from the management. However, they were quite sceptical and did not want to wet their feet (i.e., they wanted it to be a bottom-up approach). They rather wanted the change to use very little or no resources. **Team Commitment** *(score 2)*. The team was aware of the positive results of the changes and was willing to execute them if led properly. They often did not see the long-term consequences and wanted to see effects here and now. But after a discussion and reviewing evidence they usually allowed the change. **Effort** *(score 3)*. The change took a fair amount of additional resources due to the lack of on-site knowledge and proper management support. The DICE® score in this case is 18 *(Woe zone)*. This means that this particular change initiative was carried out in a very unfriendly environment and had good chances for a disaster. Some of the factors are in fact at the developer level (i.e. team commitment), but at this point the authors recognise that this particular change process would need much more top-down support rather than bottom-up. If the change process is not heavily supported by the management, it is likely to fail. This is also in line with the DICE® formula. Differences in the perception of development methodologies deployment by managers and developers were studied by Huisman and Iivari [7].

3 Reflection, Outcome and Conclusions

Rules like good programming style and techniques do not come from managers saying *"use TDD"* or *"program loosely coupled components"*. Management support is substantial (as pointed out in the previous section), but clearly not sufficient. It is the authors' experience that good practices have to come from the developer level. People are more willing to change when they see their peers change with good effect.

To increase the chance of success, process changes need both active management support and the developers' commitment. Without these two forces, the change initiative is likely not to spread enough throughout the organisation and in effect will die on its own. The main issue is to change the attitude of the team and of the managers. As Beck says, XP *"is about social change"* [6]. This is the turning point.

The authors suggest to complement the body of XP with additional practices and principles, which should address the problems that arise in the fragile process of introducing XP. Such practices and principles are needed to shed light on what has to be done to increase the chance of success when trying to implement change. They would not be XP practices and principles, but would rather concern introducing XP. They have to be chosen very carefully. The myriads of existing organizations do have things in common, but they also differ. As the authors' commitment, they propose a principle (*Empirical Evidence*) and an accompanying practice (*Joint Engagement*).

Empirical Evidence means that it is wise to ground on empirical evidence when introducing changes. We search for evidence to confirm whether or not we are on the right track with the process change. With empirical evidence comes the notion of context in which the evidence was obtained, limitations of the empirical studies, etc.

One of the widely accepted sources of evidence on the introduction of changes is the DICE® framework. However, other sources of evidence are welcome as well.

Joint Engagement is the new practice guided by the *Empirical Evidence*, as well as the *Accepted Responsibility* [6] principles. Following the *Joint Engagement* practice we begin the change process at various levels of an organisation. The aim is to have the DICE® score in the *Win* zone. We *Win* when we are able to implement changes successfully and permanently. The Boston Consulting Group's studies of the DICE® framework proves that keeping this score low considerably raises the chance of success of the change process [3]. We monitor the DICE® score and try to keep it low. Individuals at the management and at the developer level should be educated and involved in the process. They have to willingly accept their diverse responsibilities in the change process (*Accepted Responsibility* principle). This could be done by means of e.g. the first XP project in the organisation [6]: a meta-project of implementing XP.

The new practice differs from the XP's original *Whole Team* practice in that the latter emphasises diverse team competences rather than the sensible interaction of the lower-level staff with the organization's management during process change programs. Following this new practice and principle is not very surprising for some, but the history of XP itself shows the value of naming and systematizing well-known behaviours into such concrete forms.

Changes to organisations are painful, but experience shows that if proper actions are taken, the change programs, and the introduction of agility in particular, may be very successful. The reflection presented in this paper identifies a need for a set of principles and practices that would support the introduction of agile techniques to organisations. For a start, the authors propose the duo of the *Empirical Evidence* principle and the *Joint Engagement* practice. The authors see a necessity for an assorted and explicit toolkit of introductory practices and principles to help organisations embrace change. They will further investigate this subject to extend this toolkit. Contributions from other practitioners are welcome.

References

1. Madeyski, L., Stochmiałek, M.: Architectural Design of Modern Web Applications. Foundations of Computing and Decision Sciences **30**(1) (2005) 49–60 http://madeyski.e-informatyka.pl/download/23.pdf
2. Beck, K.: Test Driven Development: By Example. Addison-Wesley (2002)
3. Sirkin, H.L., Keenan, P., Jackson, A.: The Hard Side of Change Management. Harvard Business Review **83**(10) (2005) 108–118
4. DICE Framework http://www.12manage.com/methods_bcg_dice_framework.html
5. Ziółkowski, B., Drake, G.: Rolling the DICE® for Agile Software Projects. In Abrahamsson, P., Marchesi, M., Succi, G., eds.: XP 2006. Volume 4044 of Lecture Notes in Computer Science., Springer (2006) 114–122
6. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change. 2nd edn. Addison-Wesley (2004)
7. Huisman, M., Iivari, J.: Deployment of systems development methodologies: perceptual congruence between is managers and systems developers. Inf. Manage. **43**(1) (2006) 29–49