

# Can Diversity in Global Software Development be Enhanced by Agile Software Development?

Orit Hazzan

Department of Education in Technology and Science  
Technion – Israel Institute of Technology  
Haifa, Israel  
(9724) 829 3107

oritha@tx.technion.ac.il

Yael Dubinsky

Department of Computer Science  
Technion – Israel Institute of Technology  
Haifa, Israel

yael@cs.technion.ac.il

## ABSTRACT

In this paper we discuss the notion of diversity in software development processes in general and its relevance with respect to global software development (GSD) in particular. Based on the following two working assumptions: a) diversity benefits with communities and organizations that welcome it and b) diversity is an inherited element of GSD, we discuss how agile software development, which allows for diversity, may support GSD processes. We illustrate this argument by demonstrating the expression of gender diversity, management diversity and opinion diversity in agile software development environments.

## Categories and Subject Descriptors

D.2.6 [Software Engineering]: Management – *Programming teams, Software process models.*

## General Terms

Management, Economics, Human Factors.

## Keywords

Global software development, diversity, agile software development.

## 1. INTRODUCTION

According to the CfP of this workshop "managers and developers engaged in global software development agree that cross-site, cross-cultural projects do not just happen." Rather, projects are confronted by countless challenges, from project set-up, to progress control, to day-to-day communication, and even to managing of cultural conflicts. All of these issues can become serious obstacles that require careful examination and practical solutions." Following this statement, in this paper we highlight the concept of diversity in software development environments, and question whether agile software development may help overcome these obstacles, which are inherited, by nature, in any global software development (GSD) setting.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
*Conference'06*, May 20–28, 2004, Shanghai, China.  
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

Diversity can be expressed in different ways, such as nationalities, gender, minorities, cultures, life styles and world views. Since more and more companies become global, diversity becomes an integral characteristic of software development teams; therefore diversity can not be neglected when GSD is discussed. At the same time, studies tell us that no matter what shape diversity takes, diversity benefits with societies that welcome it (Cf. for example, Florida, 2002). Therefore, and not surprisingly, diversity is perceived as a powerful management practice (see for example, Toyota's 21<sup>st</sup> Century Diversity Strategy<sup>1</sup> and David Thomas's paper in *Harvard Business Review*<sup>2</sup>). In the same spirit, diversity is introduced also into agile and lean software development processes. Here are several examples: Adaptive Software Development (Highsmith, 1999), Scrum (Rising and Janoff, 2000), Extreme programming (Beck with Andres, 2005) and Crystal Orange (Cockburn, 2002). For illustration we quote Beck (2005):

Teams need to bring together a variety of skills, attitudes, and perspectives to see problems and pitfalls, to think of multiple ways to solve problems, and to implement the solutions. Teams need diversity. (p. 29).

Accordingly, we suggest that dealing with diversity is one of the key elements of software development processes in general and of GSD in particular.

There are many ways by which we could approach diversity in this paper. Among them, we could discuss the benefits of diversity or illustrate case studies from other fields. In this position paper, however, we employ a different approach and present data, taken from two of our research works conducted in agile software development environments, that illustrate how agile software development naturally foster diversity. As will be illustrated, this paper addresses the following topics listed in the workshop CfP: Software engineering methodologies & processes for GSD, Organizational models and strategies, Project management, Managing peopleware (e.g., cross-cultural conflicts, distributed collaboration), Knowledge management, Software process and practice improvement, Education and training of practitioners, Agility and global software development, and Team communication and coordination.

<sup>1</sup> Toyota's 21<sup>st</sup> Century Diversity Strategy:

<http://www.toyota.com/about/diversity/21stcenturyplan.pdf>

<sup>2</sup> David Thomas, "Diversity as Strategy", *Harvard Business Review*, September 2004, pp. 98-108.  
<http://www.gpworldwide.com/quick/sep2004/art2.asp>

This position paper is based on our intensive experience in introducing agile software development both in academia and in industry (cf. for example, Hazzan and Dubinsky, 2003; Dubinsky and Hazzan, 2005, Tomaylo and Hazzan, 2004, and our website [http://edu.technion.ac.il/Courses/cs\\_methods/eXtremeProgramming/XP\\_Technion.htm](http://edu.technion.ac.il/Courses/cs_methods/eXtremeProgramming/XP_Technion.htm)).

In section 2 we present the main ideas of agile software development. In Section 3 we illustrate how several agile software development practices enhance diversity. Specifically, we illustrate how the culture inspired by agile software development environments may create a diversified development environment that can be expressed in different terms. Among them, we illustrate gender diversity, management diversity and opinion diversity.

## 2. AGILE SOFTWARE DEVELOPMENT

During the 1990's, the agile approach towards software development started to emerge in response to problems in the software industry. Specifically, the agile software development approach, composed of several methods, formalizes software development frameworks which aim at overcoming characteristic problems of software projects (Highsmith, 2002). The Manifesto for Agile Software Development (<http://agilemanifesto.org/>) appears in Table 1.

**Table 1: Manifesto for Agile Software Development**

<p>We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:</p> <ul style="list-style-type: none"> <li>▫ <b>Individuals and interactions</b> over processes and tools</li> <li>▫ <b>Working software</b> over comprehensive documentation</li> <li>▫ <b>Customer collaboration</b> over contract negotiation</li> <li>▫ <b>Responding to change</b> over following a plan</li> </ul> <p>That is, while there is value in the items on the right, we value the items on the left more.</p>
--

The agile approach reflects the notion that software development environments should support communication and information sharing, in addition to heavy testing, short releases, customer satisfaction, and sustainable work-pace for all individuals involved.

In fact, agile software development is about cultural changes in software development environments. Among other cultural changes that agile software development fosters, we address in this paper the cultural change that allows for diversity. In other words, based on our assertion that diversity is important for software development environments, we suggest that it is not sufficient to call for diversity and to wait that it emerges; rather, a cultural change that allows for diversity should take place. We propose that the agile approach towards software development embraces cultural change that one of its aspects inspires diversity.

For example, the planning game is one of many activities that take place in agile software development environments that allows for diversity. The planning game is an activity by which the details of the actual development process are determined. In this activity all the parties involved in the software development – development, business, management and customers – participate. Naturally, such a setting allows for diversity since everyone can express

his/her opinion and point of view. This idea is further elaborated in Section 3.3.

## 3. ILLUSTRATIONS: AGILE SOFTWARE DEVELOPMENT ENHANCEMENT OF DIVERSITY

### 3.1 Gender Diversity

Recent managerial research studies have identified specific characteristics that are attributed to “women’s management style”. Here are two examples from the literature (*italics* ours).

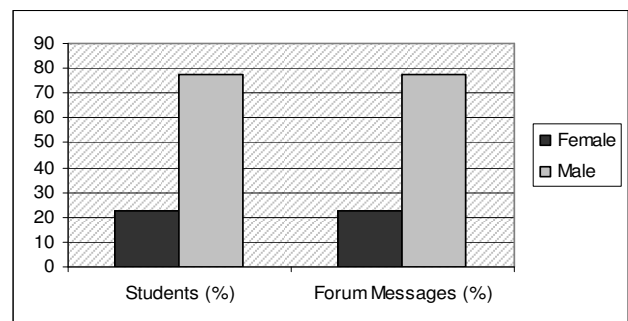
- “Women’s style of management is based on sharing power, on inclusion, consultation, consensus, and *collaboration*. Women work *interactively and swap information* more freely than men do. Women managers encourage their employees by *listening to, supporting, and encouraging* them.” (Fisher, 1999)
- “Recent research indicates women’s management style, which is centered on *communication* and *building positive relationships*, is well suited to the leadership paradigm of the 90’s.” (Peters, 2003)

We note that male managers employ these strategies as well. Nevertheless, such quotes clearly reflected that this management style is similar to the perspective of the agile software development approach. For example, both approaches convey collaboration and communication.

Indeed, as the following data show, agile software development environments can inspire gender equal participation. This data was gathered by observing a project-based operating-systems course that the second author instructs at the Computer Science Department of the Technion - Israel Institute of Technology (for more details see Dubinsky and Hazzan, 2005). In this course, the agile method has been used since the 2002 Summer Semester by four teams of 10-12 students each semester. Each team is guided by an academic coach.

An examination of the communicative behavior of 294 students who worked according to the agile method during eight semesters in 27 different groups reveals that females are as communicative as males in this setting.

**Figure 1: Females and males’ communicative behavior in agile teams**



For example, when communicative behavior was measured by monitoring the electronic forum used by students in the course (See Figure 1), it was observed that the percentage of messages sent by females (22.8% or 1391 out of 6093) was essentially the

same as the percentage of females in the cohort (22.4% or 66 out of 294). In other words, these data illustrate how the agile approach towards software development creates an environment in which women and men communicate similarly.<sup>3</sup>

### 3.2 Management Diversity

Agile software development in general and Extreme Programming (XP) in particular advocates the idea of 'whole team' which suggests that "a variety of people work together in interlinking ways to make a project more effective. They have to work together as a group for each to be successful." (Beck with Andres, 2005, p. 73). We apply this idea also with respect to the management of general agile software development projects. Specifically, in the teams with which we work, in addition to being a software developer, each team member plays an auxiliary role on their team. Our role scheme is presented in detail in Dubinsky and Hazzan, 2005. Table 2 presents the role scheme.

**Table 2: Roles in Agile Team (Dubinsky and Hazzan, 2005)**

Group of Roles	Role	Role Description
Leading Group	Coach	Coordinates and solves group problems, checks the web forum and responds on a daily basis, leads some development sessions.
	Tracker	Manages the group diary, measures group progress with respect to the estimations and test scores, manages and updates the boards.
	Methodologist	Learns the software development method applied in the course and guides the software process when required, guides and supports the other team members with their role.
Customer Group	Customer	Tells customer stories, makes decisions pertaining to each iteration, provides feedback, defines and develops acceptance tests.
	Acceptance tester	Works with the customer to define and develop acceptance tests, learns the topic of test-driven development and instructs it to the other team members.
Maintenance Group	Presenter	Plans, organizes and presents version presentations, demos, and time schedule allocations.
	Documenter	Plans, organizes and presents the project documentation: process documentation, user's guide, and installation instructions.
	Installer	Plans and develops an automated installation kit, supports and instructs other teammates as to the appropriate way to develop software for easy and correct installation.

<sup>3</sup> We do not argue that other software development approaches do not allow for diversity. At the same time, however, it is accepted that diversity lacks in many software teams.

Code Group	Designer	Maintains current design, works to simplify design, searches for locations in the software that need refactoring and ensures proper execution of such.
	Code reviewer	Establishes and refines group code standards, searches for development tools that can help the team, guides pair programming, guides and supports in the maintaining of the code standards and tools.
	Unit tester	Learns about unit testing, establishes an automated test suite, guides and supports other team members in the development of unit tests.
	Integrator	Establishes an integration environment including source control, publishes rules pertaining to the addition of new code using the test suite, guides and supports other teammates in the integration task.

As it turns out, the concept of whole team together with the role scheme not only enhance careful management of each aspect of the software project, but further, at the same time, allow for the integration of a variety of approaches towards the project management; in other words, the combination of these two notions enhances diversity.

To illustrate this idea, following are expressions of role holders taken from our research in the academia (*italics* ours).

- "1. Developing in a team allows the project goal to be more challenging and much more interesting. 2. *In this way you can learn more about other people ideas and solutions.*"
- "[It] taught me personally *how to cope with other people's shortcomings* as well as on my own, and *interact well with team members despite some disagreements.*"

An examination of the expansion of this role scheme to a long-term software project, in which roles can be carried out by different team members during the development process, clarifies how, by enabling each team member to view the development process from different perspectives and to improve his or her understanding of the others' perspectives, this role scheme enhances diversity.

The generalization to GDS processes is immediate. On the individual level the role scheme increases one's awareness to the different viewpoints expressed during a development process; on the team level, the entire team has the opportunity to view the development process from different perspectives related to the different roles.

### 3.3 Opinion Diversity

This kind of diversity is highlighted by the practice of short cycles which, in agile software development processes implies that the developers, as well as the customers, have an on going opportunity both to improve their understanding of the developed product and to express their perspectives. One of these opportunities takes place during the planning game, mentioned above. Another opportunity is the daily stand up meetings.

In what follows we demonstrate how in agile software development environments a diversified perspective can be

expressed and reconsidered during the entire software development process. This situation contrasts plan-driven software development processes in which the output of each stage can hardly be changed as soon as it is transferred to the next development stage.

In this case we use data from our research at the software unit of the Israeli Air Force (Dubinsky, Hazzan, Talby, and Keren, in press, 2006), in which the system analyst role was examined when the development process was changed from a plan-driven method to an agile one.

During the change process, system analysts, who previously worked in a separate group that analyzed the customer requirements and delivered a detailed specifications to the developers, currently, after the change took place and the agile method XP (Beck with Andres 2005) has been applied, work in an hybrid way. In this new development environment, *general* specifications are written by a separate group which works with the customer, and the detailed specifications are written by the 'whole team' – the team of system analysts together with the developers, customer and testers – each two-week iteration. Following are their expressions after six months of transition to agile (*italics* ours):

- “*Everyone is involved* and this raises the confidence feeling with respect to the process.”
- “There is *more interaction*.”
- “*Explaining the concept, I sometimes see that my concept is wrong*”.

Further, our data illustrate that diversity encourages opinion expression by all players, as is expressed in the following quotes:

- Customer: “Every two weeks I need to say what will be in the iteration.”
- Team leader: “I *constantly search for subjects to raise* in our two-weekly session of planning game.”

#### 4. CONCLUSIONS

The working assumption of this paper is that diversity, no matter how it is expressed, benefits with software development processes. Based on this working assumption we aimed at conveying the idea that since agile software development environments foster diversity, it has the potential to support GSD projects, in which diversity is a built-in element and should be coped with anyway. Based on our research experience in teaching, guiding and researching agile software development processes, we

illustrated our ideas by three kinds of diversity: gender diversity, management diversity and opinion diversity.

#### 5. REFERENCES

- [1] Beck, K. with Andres, C. (2005). *Extreme Programming Explained*, 2<sup>nd</sup> edition. Addison-Wesley.
- [2] Cockburn, A. (2002). *Agile Software Development*, Addison-Wesley.
- [3] Dubinsky, Y. and Hazzan, O. (2005). A framework for teaching software development methods, *Computer Science Education* **15**(4), pp. 275-296.
- [4] Dubinsky Y., Hazzan O., Talby D., and Keren A. (in press, 2006). System Analysis and Design in a Large-Scale Software Project: The Case of Transition to Agile Development, *8th International Conference on Enterprise Information Systems (ICEIS)*, 23 - 27 May 2006, Paphos, Cyprus.
- [5] Fisher, H. (1999). *The First Sex – The Natural Talents of Women and How They are Changing the World*. Ballantine Books, NY.
- [6] Florida, R. (2002). *The Rise of the Creative Class*, Basic Books.
- [7] Hazzan, O. and Dubinsky, Y. (2003). Teaching a Software Development Methodology: The Case of Extreme Programming, *Proceedings of the 16<sup>th</sup> International Conference on Software Engineering Education and Training*, Madrid, Spain, pp. 176-184.
- [8] Highsmith, J. A. (1999). *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, Dorset House Publishing Company.
- [9] Highsmith, J. (2002) *Agile Software developments Ecosystems*, Addison-Wesley.
- [10] Peters, H. (2003). Risk, rescue and righteousness: How women prevent themselves from breaking through the glass ceiling, Hagberg Consulting Group.
- [11] Rising, L. and Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams, *IEEE Software*, July/August 2000, pp. 26-32.
- [12] Tomayko, J. and Hazzan, O. (2004). *Human Aspects of Software Engineering*, Charles River Media.