

Promoting Natural Human Attitude Towards Work: Scrum

Podrška prirodnom odnosu ljudi prema radu: Scrum

Valentino Vranić

Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava
Faculty of Business Valjevo, Singidunum University

Apstrakt – Prirodan odnos ljudi prema radu, koji se ispoljava kroz njihove osobine kao što su potreba za kreativnošću, nesposobnost poimanja udaljenih termina, prilagodavanje mehanizmima evaluacije njihovog rada, potreba za ličnim kontaktom i komunikacijom, potreba da vide i prezentiraju rezultate svog rada ili averzija prema spoljašnjem upravljanju, se obično iz perspektive direktivnog menadžmenta uzima kao problematičan. S druge strane, potiskivanje prirodnog odnosa ljudi prema radu vodi ka demotivaciji i smanjenju kreativnosti, što ugrožava opstanak u današnjem svetu narastajuće konkurencije. Umesto ovoga, Scrum kao okvir za realizaciju projekata, uspeva da izvuče korist iz ljudskih osobina uzdržavajući se od direktivnog menadžmenta. Lični kontakt i komunikacija igraju fundamentalnu ulogu u Scrumu oživljavajući ceo proces razvoja. Učenje na greškama i izvođenje retrospektive zauzima posebno mesto u Scrumu u saglasnosti s neprestanom ljudskom potrebom za učenjem i usavršavanjem.

Ključne reči - software development, technology, people, work, Scrum.

Abstract – Natural human attitude towards work manifested through their habits such as need for creativity, inability to comprehend distant deadlines, adapting to mechanisms of evaluation of their work, need for personal contact and communication, need to see and present results of their work, or aversion towards outer control, is usually perceived as problematic from the perspective of directive management. On the other hand, suppressing natural human attitude towards work leads to demotivation and decrease in creativity, which jeopardizes survival in today's increasingly competitive world. Instead of this, Scrum as a framework for project realization manages, to benefit from these human habits by refraining from directive management. Personal contact and communication play a fundamental role in Scrum making the whole development process live. Learning from mistakes and making retrospective takes a special place in Scrum complying with constant human need for learning and improvement.

Index terms – software development, technology, people, work, Scrum.

1. INTRODUCTION

Technology is the key notion of today's software development scene. However, technology is invented by the people for the people and should adjust accordingly reflecting people's needs. One of these needs is the need to work, which may appear as its opposite—i.e., the need *not* to work—from the perspective of what is commonly perceived as organized work.

Modern age exhibits constant need for newer and better products. Quantity is being successfully addressed by automation, but product improvement also requires creativity, and that can't be automated.

Technology is commonly thought of as being tightly related to mass production and work specialization. While having a big quantity of products produced in a short time and by highly specialized workers that know their job well seems pleasing at first sight, who would really like to stand at an assembly line? How would an assembly line worker adapt to new requirements?

The technology as a word almost bears a sound of machines, tools, or devices in it. However, its origin is different from that. Greek τέχνη (techne) stands for art and skill [6], or craft [5], all with essential creativity in their meaning. How not to make software technologies be assembly lines or—more precisely—how not to make assembly line workers out of software developers?

This paper explores how can the natural human attitude towards work fit into this milieu in a constructive manner in the area of software development. Section 2 recalls how ideas are changed into reality in software development. Section 3 explains how human habits seemingly contradict to organized work. Section 4 debunks this cliché by showing how organized work can benefit from these human habits in Scrum, a popular agile framework for project realization. Section 5 points to related work, and Section 6 concludes the paper.

2. PROJECTING IDEAS INTO REALITY

An organized effort to fulfill an idea is commonly denoted as project. A software development project comprises (requirements) specification, analysis, design, and implementation, followed by integration, testing, and maintenance. These typical phases—or better to say activities—can be identified in projects in general; however, here we will focus on software development projects.

Ideally, we would like to get specified what to do, design it, and implement it, flowing smoothly towards a successful project end—like a waterfall. In practice, the requirements are often underspecified, ambiguous, inconsistent or contradictory, or overspecified. Underspecification is often a consequence of failing to capture some requirements, which can be corrected. However, the specification is also being incomplete with respect to what the client *would* want to get at the end. This differs from what the client wanted to get in the beginning. The client's ideas change, and the software being developed influences this, too.

This problem is colloquially known as “the client doesn't know what he wants,” but the real issue underneath is that it's not possible to make a complete specification before its realization, i.e. before the software is developed. With iterative and incremental development—especially such that the client gets a functional product after each iteration—a natural point where changes can be introduced occurs: in between two iterations. Change of requirements becomes a means of project control in hands of the client.

3. PEOPLE AND PROJECTS

How do people deal with work to be done in projects and how this conforms to common expectations? It seems these are quite contradictory. People make mistakes and learn from their mistakes. But we want skilled professionals who know their job and make no mistakes.

To people, monotonous and dull work is hard to endure. But we need the work to be done and it's not their job to think about it.

People like working on interesting things. But the most of the work is boring and there's nothing that can be done with that.

People can really understand only sufficiently close deadlines. But we must plan far ahead if we want to get to our goal.

People adapt to the mechanisms of evaluation of their work. But we have to calculate what they are worth somehow, and we can't stand behind their shoulders all the time.

People need personal contact and direct communication with other people. But the communication takes a lot of time and easily gets informal, while the real work is waiting to be done.

People need to see the results of their work and need to expose them to other people. But there's no time to involve all participants in the whole picture.

People don't like outer control. But without outer control they wouldn't work.

Classical approach to project realization or organized production in general is in a direct opposition to human habits. Its success is their suppression, and that's no success at all. The result is a group of obedient—but demotivated—people that have adapted their work to the mechanisms of evaluation. This is called management. Is it possible to avoid this and yet to profit out of the natural human attitude towards work?

4. SCRUM

The other way around resolving the opposition of human habits and management is to accept human habits and refrain from directive management. This is what's essentially being done in Scrum, where entrusted team of developers with all members being equal in their rights (there's no leader) delivers high quality work on time. Scrum is a framework for non-directive project realization. It's often associated with software development, but it has been successfully applied in many other areas [8].

4.1 No Project Manager

What has to be done is represented by a devoted role out of the (development) team called Product Owner. This way it is a live person the team can talk to about requirements, and not just a dead specification document. Product Owner is responsible for tracking and ordering undone work in the product backlog to ensure working on most important things at all times as the team may take work always from the top of the product backlog.

While the team leader role would be counterproductive, a coach, guide, or advisor (but not a supervisor) is beneficial. This is ScrumMaster, whose task is to protect team, remove any impediments, and facilitate Scrum in the team. There's no project manager.

4.2 Communication and Retrospective

In Scrum, direct communication is fostered by regular meetings, but—even more importantly—it is a means of cooperation among the team members. Everything is made visible: the tasks, progress, failures... The team is small enough (three to seven members) to let every member get involved and even more: committed to work.

It's very important to take time and analyze what in the project is good, and what could be improved. This kind of retrospective is made explicit in Scrum and happens regularly after each sprint.

In Scrum, mistakes are considered to be a driving force of process improvement. Of course, nobody wants to make mistakes, but if they happen in Scrum, they are taken positively: as an opportunity for improvement. Each mistake is analyzed how and why it has been made and what to do to avoid similar mistakes in future.

Learning from mistakes and making retrospective demonstrate how Scrum incorporates learning. A Scrum-like approach called Open Agile makes learning explicit activity [9]. Scrum is even being employed in education as such. A successful application has been reported at the secondary school level [10] and university level [11].

4.3 Iterations and Planning

The work is performed in a week to four weeks long iterations called sprints, each ending in a potentially shippable product demonstrated at the end of each sprint to the client and all interested. The deadline a few weeks ahead is readily comprehensible to people. Keeping all sprints equally long makes team establish a certain rhythm or pulse of development.

The team alone determines how much work they will take into the next sprint. The concern about sprint failure prevents the team from taking too much work, while taking more appears as a challenge to them, so they naturally seek for balance. By this, the team works at a certain speed as if it would be running (sprint), taking care of not becoming exhausted (the work is to be performed in normal working hours), but also of not turning running into walking, i.e. not becoming too slow.

The client determines what is to be done and can change that over time and is welcome to do so by initiating a change of the project backlog in between sprints.

Scrum is frank with planning: it is based on an effort estimate. In addition, this estimate is relative: expressed in terms of how much one issue requires effort compared to another one. Effort is expressed in points that gain their real meaning with respect to work in sprints.

5. RELATED WORK

Patterns describe how to resolve conflicting forces [1]. Contradictory statements in Sect. 3 can be viewed as such forces. This is similar to organizational patterns [3] that may be used to describe Scrum [4].

Scrum practitioners report Scrum requires increased discipline and diligence compared to directive approaches [2]. Here, Scrum has been found to comply with not quite praised natural human habits. Somewhat paradoxically, when not suppressed, but instead given freedom, natural human habits and natural human attitude towards work seem to wake—also natural—self-discipline. Interestingly, this is conformant to the findings of Maria Montessori with respect to childhood education [7].

6. CONCLUSIONS

The main obstacle of the waterfall project model in software development is that the software being developed affects the client's idea of it. By making partial results visible to the client, iterative and incremental development alleviates this conflict and also provides a better environment for natural human attitude towards work such as need for creativity, inability to comprehend distant deadlines, adapting to mechanisms of evaluation of their work, need for personal contact and communication, need to see and present results of their work, or aversion towards outer control.

Suppressing natural human attitude towards work leads to demotivation and decrease in creativity, which jeopardizes survival in today's increasingly competitive world. Instead of this, Scrum as a framework for project realization, manages to benefit from these human habits by refraining from directive management. Personal contact and communication play a fundamental role in Scrum making the whole development process live. Learning from mistakes and making retrospective takes a special place in Scrum complying with constant human need for learning and improvement.

Acknowledgment. The work was supported by the Scientific Grant Agency of Slovak Republic (VEGA) grant No. VG 1/1221/12 and by the Slovak Research and Development Agency under the contract No. APVV-0233-10.

REFERENCES

- [1.] C. Alexander. *The Timeless Way of Building*. University of Oxford Press, 1979.
- [2.] M. Bauer. How to Succeed with Scrum. 2011. <http://www.martinbauer.com/Articles/How-to-Succeed-with-Scrum>
- [3.] J. O. Coplien and N. B. Harrison. *Organizational Patterns of Agile Software Development*. Prentice Hall, 2004.
- [4.] Gertrud&Cope. Scrum as Organizational Patterns, 2011. <https://sites.google.com/a/scrumorgpatterns.com/www/>
- [5.] LEXIKOLOGOS, Modern Greek Dictionary. http://www.lexilogos.com/english/greek_dictionary.htm
- [6.] Merriam-Webster Dictionary. <http://m-w.com/>
- [7.] M. Montessori. *Upijajući um*. DN Centar, second edition, 2006. (In Serbian; original title: *La mente del bambino (Mente assorbente)*; *The Absorbent Mind*.)
- [8.] J. Sutherland. *Jeff Sutherland's Scrum Handbook*. Scrum Training Institute Press, 2010. <http://jeffsutherland.com/scrumhandbook.pdf>
- [9.] OpenAgile. The OpenAgile Primer. Version 1.1. 2011. <http://www.openagile.com/>
- [10.] Scrum: The Future for Education? Jeff Sutherland's Scrum Log, 2012. <http://scrum.jeffsutherland.com/2012/04/scrum-future-for-education.html>
- [11.] M. H. H. Willeke. Agile in Academics: Applying Agile to Instructional Design. *Proc. of Agile2011 Conference*, IEEE, Salt Lake City, USA, 2011.