Comparing Scrum and XP Agile Methodologies Using Dynamic Simulation Modeling

Nasiopoulos K. Alexandros, Damianos P. Sakas, D.S. Vlachos and Nasiopoulos K. Dimitrios

Introduction

It has been fifteen years since the articulation of agile manifesto in 2001, which brought great changes in software application development (Dingsøyr et al. 2012). According to the Manifesto for Agile Software Development (agileAlliance.org), agile methods value "(1) individuals and interactions over processes and tools, (2) working software over comprehensive documentation, (3) customer collaboration over contract negotiation, and (4) responding to change over following a plan" (Wadhwa and Sharma 2015; Salo and Abrahamsson 2007; Mushtaq and Qureshi 2012).

Agility is the ability to detect and address the business perspective to remain inventive and aggressive in a labile and rapidly changing business environment. The continuous evolution of twenty-first century technology forced companies' environment to become increasingly dynamic and organizations to constantly modify their software requirements to adjust with the new environment (Moniruzzaman and Hossain 2013).

Scrum Methodology

Scrum methodology is a method that tries to keep things simple in a constantly shifting business environment. Scrum is composed of short, strenuous daily meetings of the project team, in order to deliver quality software in 24-h short-time periods called "sprints".

N.K. Alexandros (✉) · D.P. Sakas · D.S. Vlachos · N.K. Dimitrios
Department of Informatics and Telecommunications, University of Peloponnese,
Tripoli, Greece
e-mail: pcst15014@uop.gr

A. Kavoura et al. (eds.), *Strategic Innovative Marketing*, Springer Proceedings
in Business and Economics, DOI 10.1007/978-3-319-56288-9_52

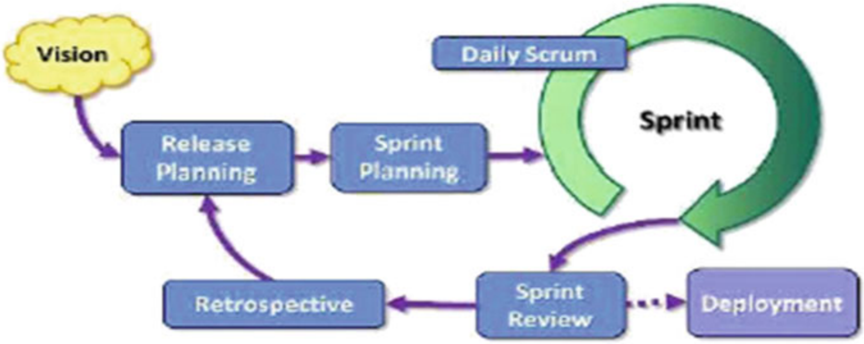


Fig. 1 Scrum methodology process flow (Mahalakshmi et al. 2013)

The key principles of Scrum methodology are the following:

- Small working teams for better communication and less expenses;
- Adaptability to changes in order to produce quality software;
- Daily software “builds” that can be tested, documented and used for further implementation;
- Distinct work segmentation and assignment to teams;
- Constant documentation and testing of the produced product;
- Ability to characterize a product as “finished” whenever required (Yadav 2015) (Fig. 1)

XP Methodology

XP methodology focuses on the constant interaction of customer, manager, and programmer and clearly defines the role each one has. In XP methodology small releases are produced periodically and tested, in order to maintain customer satisfaction through the life cycle of the software development (Fig. 2).

The key principles of XP methodology are as follows:

- Customers should participate actively in the whole process;
- Small releases are produced periodically and tested to gain early feedback from the customers;
- XP team is planning the work for the next release to reach the goals from the customers, within specific time and funds;
- Each member of the team must have full acknowledgment of how the entire product works and own the skills to improve it;
- Code must be continuously inspected for simplicity, refactoring, and tested for integration and errors (Yadav 2015).

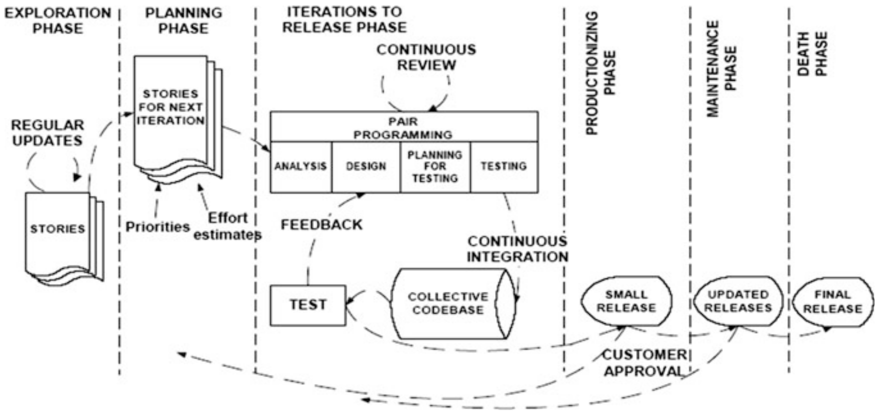


Fig. 2 XP methodology process flow (Al-Saleem et al. 2015)

In the next chapters we will proceed to the simulation and modeling of the two procedures, in order to compare them for finding the most appropriate to be applied in the case of collaboration tools.

Dynamic Simulation Model System Analysis

Since we analyzed the key principles of each methodology, we can create diagrams and tables dynamically (Richardson 2013; Sakas et al. 2014). To achieve that we made use of iThink software, by isee^{TM} systems, using this software, we can simulate the influence of each factor to the system, as time pass by (Sakas et al. 2014; Jansen-Vullers et al. 2006).

The available company resources for this project, represented by “Company Resources” stock, are moved to “Working Teams”, through “CR2WT” flow (Fig. 3). Then they are shared through flows and converters to “Work Segmentation” and “Changing Adaptability”. The distribution of resources has been chosen so as to provide the best results to the procedure of implementing the collaboration tool. After satisfaction of these factors occurs, the lamps become green, the resources of “Work Segmentation” and “Changing Adaptability” are gathered to the “Daily Software”. The next stage is the procedure of “Documentation and Testing” which after being satisfied leads to the software development (Fig. 4).

Figure 5 represents the graphical results for 12 months for five stocks, “Satisfaction Work Segmentation”, “Satisfaction Changing Adaptability”, “Satisfaction Documentation and Testing”, “Satisfaction Product”, and “Company Resources”.

When simulating the XP methodology, “Company Resources” are moved to the “Team”, through “CR2T” flow and are distributed to the “Team Regroup” and “Release” (Fig. 6). “Customers” also give resources to the “Team Regroup” and

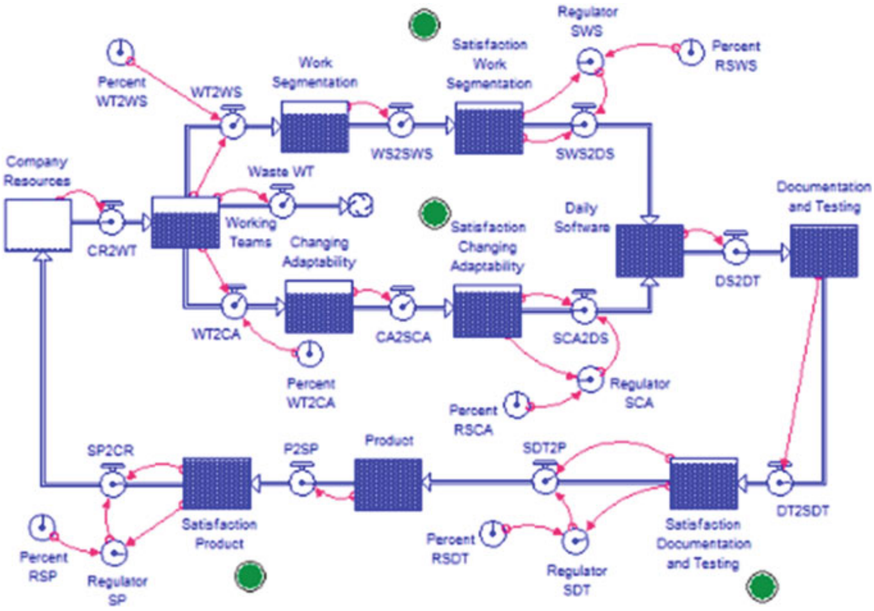
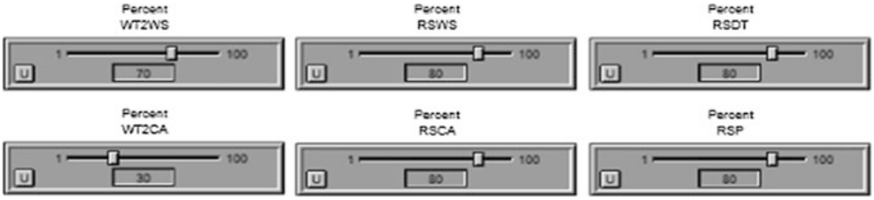


Fig. 3 Scrum methodology dynamic simulation model



Percent WT2WS: 1 — 100, U, 70

Percent RSWS: 1 — 100, U, 80

Percent RSDT: 1 — 100, U, 80

Percent WT2CA: 1 — 100, U, 30

Percent RSCA: 1 — 100, U, 80

Percent RSP: 1 — 100, U, 80

Fig. 4 Scrum methodology simulation converters percentages

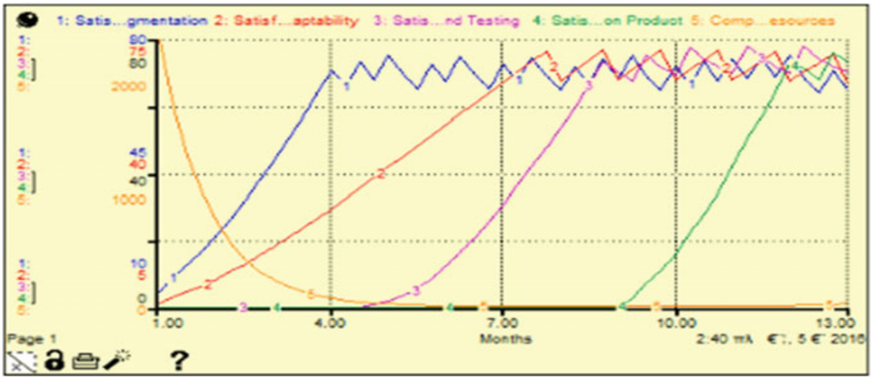


Fig. 5 Scrum methodology simulation graphical results

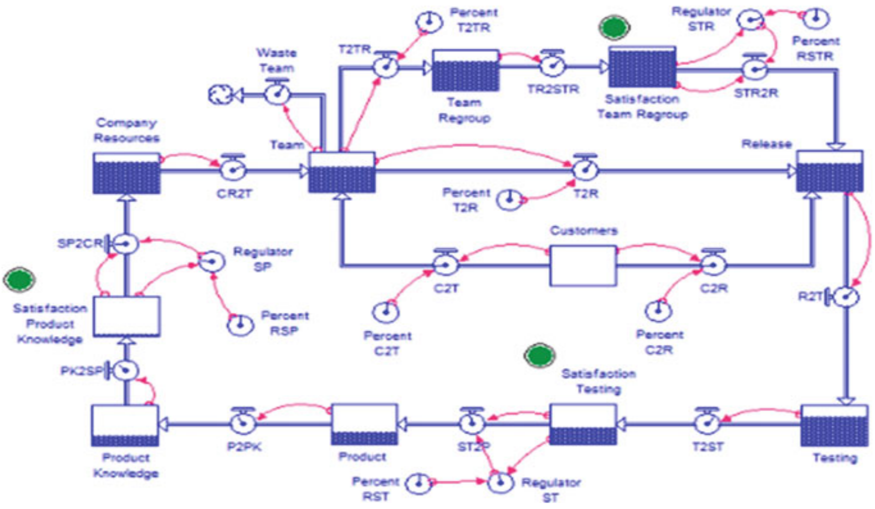
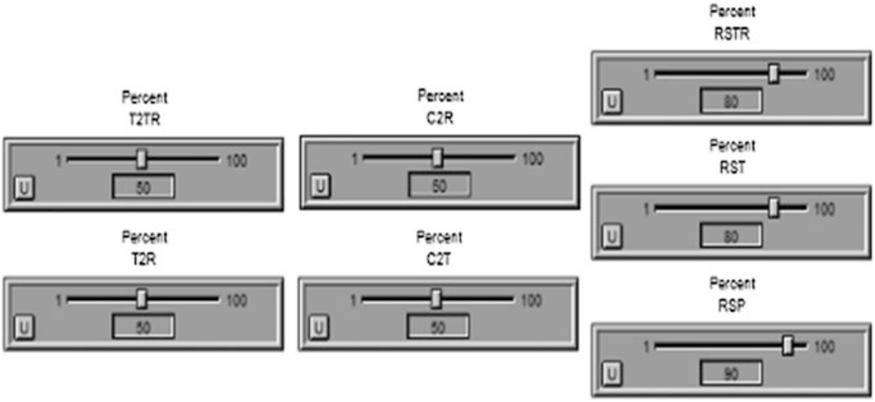


Fig. 6 XP methodology dynamic simulation model


Percent T2TR
1 100
U 50

Percent C2R
1 100
U 50

Percent RSTR
1 100
U 80

Percent RST
1 100
U 80

Percent T2R
1 100
U 50

Percent C2T
1 100
U 50

Percent RSP
1 100
U 90

Fig. 7 XP methodology simulation converters percentages

"Release", according to the principles of XP methodology. After "Satisfaction of Team Regroup" occurs, resources are also given to "Release". Next follows "Testing", "Satisfaction of Testing", and the software development in "Product". Then the team acquires knowledge of the product and "Satisfaction Product Knowledge" occurs.

Figure 7 represents the graphical results for 12 months for four stocks, "Satisfaction Team Regroup", "Satisfaction Testing", "Satisfaction Product Knowledge", and "Company Resources" (Fig. 8).

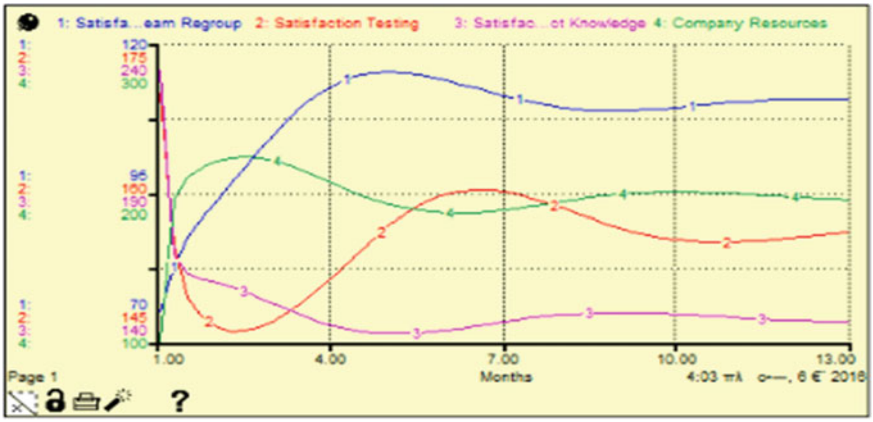


Fig. 8 XP methodology simulation graphical results

Conclusions

The comparison of the results from the simulation modeling of Scrum and XP methodologies shows that the outcomes of the use of Scrum methodology are superior in the case of developing collaboration tools. From the comparison of the two models, it appears that, in XP methodology, we do not achieve methodologies' factors satisfaction, for the case of collaboration tools.

The evolution of twenty-first century technology brought great changes to both software and the tools that software companies need to develop, in order to be competitive in a rapid changing environment (Buur et al. 2011). It is imperative to test software development methodologies in order to check their appropriateness for these new demands.

References

Al-Saleem, S.M., and H. Ullah 2015. A comparative analysis and evaluation of different agile software development methodologies. *International Journal of Computer Science and Network Security*. 15 (7).

Buur, J., and R. Mitchell. 2011. The business modelling lab. In *Participatory innovation conference.*

Dingsøyr, T., S. Nerur, V. Balijepally, and N.B. Moe. 2012. A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*. 85 (6): 1213–1221.

Jansen-Vullers, M.H., and M. Netjes. 2006. Business process simulation—a tool survey. In *Proceedings of the seventh workshop and tutorial on practical use of coloured petri nets and the CPN tools.* University of Aarhus, Denmark.

Mahalakshmi1, M., and M. Sundararajan. 2013. Traditional SDLC vs scrum methodology—a comparative study . *International Journal of Emerging Technology and Advanced Engineering* 3 (6).

Moniruzzaman, A.B.M., and S.A. Hossain. 2013. Comparative study on agile software development methodologies. *Global Journal of Computer Science and Technology*. 13 (7) Version I.

Mushtaq, Z., and R.J. Qureshi. 2012. Novel hybrid model: Integrating scrum and XP. *Information Technology and Computer Science* 6: 39–44.

Richardson, G. 2013. System dynamics. *Encyclopedia of Operations Research and Management Science* 1519–1522.

Sakas, D.P., D.S. Vlachos, and D.K. Nasiopoulos. 2014. Modelling strategic management for the development of competitive advantage, based on technology. *Journal of Systems and Information Technology* 16 (3): 187–209.

Salo, O., and P. Abrahamsson. 2007. An iterative improvement process for agile software development. *Software Process: Improvement and Practice* 12 (1): 81–100.

Wadhwa, M., and N. Sharma. 2015. Review of agile software development methodologies. *Advances in Computer Science and Information Technology.* 2 (4): 370–374.

Yadav, A. 2015. AGILE: Software development model. *International Journal of Engineering and Technical Research* 3 (3).