Agile Software Teams Can Use Conflict to Create a Better Products

Broderick Crawford[1,2], Ricardo Soto[1,3], Claudio León de la Barra[1], Kathleen Crawford[1], and Eduardo Olguín[4]

[1] Pontificia Universidad Católica de Valparaíso, Chile
[2] Universidad Finis Terrae, Chile
[3] Universidad Autónoma de Chile, Chile
[4] Universidad San Sebastián, Chile
{broderick.crawford,ricardo.soto,claudio.leondelabarra}@ucv.cl,
kathleen.crawford.a@mail.pucv.cl,
eduardo.olguin@uss.cl

Abstract. Agile Software processes emphasize collaboration more than traditional methods. Collaborations and interactions are cited directly in two of the four values listed in the agile manifesto. Because of everything that involves communication contains the potential for conflict, we are interested in knowing how to manage conflicts to enhance agile projects.

Keywords: Software Engineering, Agile Development, Conflict Management.

1 Introduction

Some amount of conflict is required in a team to motivate change and encourage creativity. The relation between conflict and team effectiveness approximates an invert U-curve, where too much or too little conflict impacts negatively. Therefore, management to maintain an appropiate level of conflict is crucial for optimizing team effectiveness.

Conflict is defined as an interactive process manifested in incompatibility, disagreement or dissonance within or between social entities (individual, group, organization, ...) [26]. In recent years the concept has taken an unexpected turn around its qualities, being discovered that the appearance of conflict in small amounts could bring great benefits to organizations in terms of creativity and innovation [21,23,13,18]. Seeing that the conflict is necessary to a creative process like software development.

2 The Conflict within Organizations

We can find many ways to classify the conflict within organizations: interpersonal, intrapersonal, intergroup and intragroup [26]. Interpersonal conflict refers to the most basic type of conflict that is developed between two people or colleagues either by differences in personality or work style also a personal history.

C. Stephanidis (Ed.): HCII 2014 Posters, Part I, CCIS 434, pp. 24–29, 2014.

Meanwhile the conflict intra staff is one where the purpose and vision of an individual differs from the overall vision of the organization. When an individual is faced with a group and is unwilling or unable to conform to the dynamic, he left the team due to an intragroup conflict. In intergroup conflict meanwhile, two teams are involved in the problem, threatening the successful completion of a project due to differences in group dynamics.

2.1 Dysfunctional and Functional Conflict

Conflict must be managed, not only as a way to optimize project success, but also to increase the satisfaction of project team members.

Persistent conflict complicates the management of the projects, causes participants to constantly disagree with each other about requirements, methods, techniques and solutions. The continued conflict damages the communication, coordination and control, reducing the team performance level affecting the final quality of the product, the project deadline accomplishment and costing. In this way we are talking about dysfunctional conflict [21].

On the other hand and with the new discoveries, we see that constructive or functional conflict is considered to have beneficial effects. Where functional conflict is present, people feel free to express their opinions, beliefs, assumptions and to challenge ideas of others [21,23,13,26]. Functional conflict works like an antidote to groupthink [6], a psychological phenomenon that occurs within a team, in which the desire for harmony or conformity in the group results in correct decisions. Group members try to minimize conflict and reach a consensus decision without critical evaluation of alternative ideas or viewpoints.

In [26] Rahim describes the main characteristics of each type of conflict.

Dysfunctional Outcomes:

- Conflict may cause job stress, burnout, and dissatisfaction.
- Communication between individuals and groups may be reduced.
- A climate of distrust and suspicion can be developed.
- Relationships may be damaged.
- Job performance may be reduced.
- Resistance to change can increase.
- Organizational commitment and loyalty may be affected.

Functional Outcomes:

- Conflict may stimulate innovation, creativity, and growth.
- Organizational decision making may be improved.
- Alternative solutions to a problem may be found.
- Conflict may lead to synergistic solutions to common problems.
- Individual and group performance may be enhanced.
- Individuals and groups may be forced to search for new approaches.
- Individuals and groups may be required to articulate and clarify their positions.

3 Developing Software

Software development projects depends mostly on team performance: "Software is developed for people and by people" [19]. But surprisingly, most of software engineering research is technical and deemphasizes the human and social aspects. It is interesting to consider the new proposals of agile methodologies for software development in order to study them considering the teamwork practices. The agile principles and values have emphasized the importance of collaboration and interaction in software development. Scrum Agile Development Process [29], the most notorious competitor of eXtreme Programming XP [4] between agilists, has attained worldwide fame for its ability to increase the productivity of software teams by several magnitudes through empowering individuals, fostering a team-oriented environment, and focusing on project transparency and results.

There are recent studies reporting efforts to improve agile process. Agile software development addresses software process improvement within teams. The work in [27][24] argue for the use of diagnosis and action planning to improve teamwork in agile software development. The action planning focused on improving shared leadership, team orientation and learning. We believe that the innovation and development of new products is an interdisciplinary issue influenced by different aspects [30][25][14], we are interested in the study of the potential of new concepts and techniques to foster knowledge management, creative problem solving and conflict management in software engineering [17][12][11].

Software engineering is a knowledge intensive process that includes some aspects of knowledge management and creative problem solving in all its phases: eliciting requirements, design, construction, testing, implementation, maintenance, and project management [19]. No worker of a development project has all the knowledge required to fulfill all activities. This underlies the need for communication, collaboration and knowledge sharing support to share domain expertise between the customer and the development team [9].

The traditional approaches (often referred to as plan-driven, task-based or Tayloristic), like the waterfall model and its variances, facilitate knowledge sharing primarily through documentation. They also promote usage of role based teams and detailed plans of the entire software development life-cycle. It shifts the focus from individuals and their creative abilities to the processes themselves. In contrary, agile methods emphasise and value individuals and interactions over processes. Tayloristic methods heavily and rigorously use documentation for capturing knowledge gained in the activities of a software project life-cycle [8]. In contrast, agile methods suggest that most of the written documentation can be replaced by enhanced informal communications among team members internally and between the team and the customers with a stronger emphasis on tacit knowledge rather than explicit knowledge [5].

One possibility to improve the software development process is to design a process which can manage conflicts. In management and business, researchers have done much work about conflict management and obtained evidence that conflict must be managed to optimize projects and boosting the satisfaction of team members. The employees who had appropriate skills - negotation and

conflict resolution, problem modeling and creative problem solving, critical thinking, written and verbal communication - worked on complex, challenging jobs, and were supervised in a supportive, noncontrolling fashion, produced more creative work. Then, according to the previous ideas the use of conflict and creativity in software development is undeniable. In a few publications the importance of creativity has been investigated in all the phases of software development process [17][16][10][20] and mostly focused in the requirements engineering [28][22]. Nevertheless, the use of techniques to foster conflict management is still shortly investigated.

4 Conflict in Software Development

Conflict management in software development must certainly take into account many factors [15,31,7,2]. We can even observe the importance of individual and interpersonal conflict mentioned above, related to levels of participation, influence and personal traits such as personality and emotions. We can also see the importance of the organization and the context in which software is developed, linked to the different views in which workers are trained. This is the *organizational culture* that has a number of rules and patterns in which the work is performed daily in an organization [3,1].

One of the most important points in the development of software is communication, it is identified as one of the critical factors [32]. This is because much of the work can be developed virtually, work teams fall into problems of misunderstanding or lack of communication which demonstrates the importance of collaborative work for the creation of better products.

5 Conclusion

We believe that the latest research shows that the use of conflict within organizations brings fruitful results around creativity and innovation projects. It is important to consider new theories of conflict management within organizations, and not to forget that we are working with human emotions and relationships.

References

1. Managing conflict in software development teams: A multi-level analysis. Journal of Product Innovation Management 15, 423–435 (1998)
2. Altay, A., Kayakutlu, G., Topcu, Y.I.: Win-win match using a genetic algorithm. Applied Mathematical Modelling 34(10), 2749–2762 (2010)
3. Barki, H., Hartwick, J.: Interpersonal conflict and its management in information system development. MIS Q. 25(2), 195–228 (2001)
4. Beck, K.: Extreme programming explained: embrace change. Addison-Wesley Longman Publishing Co., USA (2000)

5. Beck, K., Beedle, M., Bennekum, A.V., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for agile software development (2001), http://agilemanifesto.org
6. BÉnabou, R.: Groupthink: Collective delusions in organizations and markets. Working Paper 14764, National Bureau of Economic Research (March 2009)
7. Carneiro, D., Novais, P., Neves, J.: Using genetic algorithms to create solutions for conflict resolution. Neurocomputing 109, 16–26 (2013); New trends on Soft Computing Models in Industrial and Environmental Applications A selection of extended and updated papers from the SOCO 2011 International Conference
8. Chau, T., Maurer, F.: Knowledge sharing in agile software teams. In: Lenski, W. (ed.) Logic versus Approximation. LNCS, vol. 3075, pp. 173–183. Springer, Heidelberg (2004)
9. Chau, T., Maurer, F., Melnik, G.: Knowledge sharing: Agile methods versus tayloristic methods. In: Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE, pp. 302–307 (2003)
10. Crawford, B., Leon de la Barra, C.: Enhancing creativity in agile software teams. In: Concas, G., Damiani, E., Scotto, M., Succi, G. (eds.) XP 2007. LNCS, vol. 4536, pp. 161–162. Springer, Heidelberg (2007)
11. Crawford, B., Leon de la Barra, C., Soto, R., Misra, S., Monfroy, E.: Knowledge management and creativity practices in software engineering. In: Liu, K., Filipe, J. (eds.) KMIS, pp. 277–280. SciTePress (2012)
12. Crawford, B., Leon de la Barra, C., Soto, R., Monfroy, E.: Agile software engineering as creative work. In: CHASE, pp. 20–26. IEEE (2012)
13. Deutsch, M.: A Theory of Co-operation and Competition. Human Relations 2(2), 129–152 (1949)
14. Fernandez-Sanz, L., Misra, S.: Analysis of cultural and gender influences on teamwork performance for software requirements analysis in multinational environments. IET Software 6(3), 167–175 (2012)
15. Gallivan, M.J.: The influence of software developers' creative style on their attitudes to and assimilation of a software process innovation. Information & Management 40(5), 443–465 (2003)
16. Glass, R.: Software creativity. Prentice-Hall, USA (1995)
17. Gu, M., Tong, X.: Towards hypotheses on creativity in software development. In: Bomarius, F., Iida, H. (eds.) PROFES 2004. LNCS, vol. 3009, pp. 47–61. Springer, Heidelberg (2004)
18. Jehn, K.: Impact of Intragroup Conflict on Effectiveness: A Multimethod Examination of the Benefits and Detriments of Conflict. UMI Dissertation Services (1992)
19. John, M., Maurer, F., Tessem, B.: Human and social factors of software engineering: workshop summary. ACM SIGSOFT Softw. Eng., Notes 30, 1–6 (2005)
20. León de la Barra, C., Crawford, B.: Fostering creativity thinking in agile software development. In: Holzinger, A. (ed.) USAB 2007. LNCS, vol. 4799, pp. 415–426. Springer, Heidelberg (2007)
21. Massey, G., Dawes, P.: Functional and Dysfunctional Conflict in the Context of Marketing and Sales. University of Wolverhampton Business School Working Papers WP009/04. Management Research Centre, Wolverhampton University Business School (2004)
22. Mich, L., Anesi, C., Berry, D.: Applying a pragmatics-based creativity-fostering technique to requirements elicitation. Requir. Eng. 10, 262–275 (2005)

23. Miron-Spektor, E., Gino, F., Argote, L.: Paradoxical frames and creative sparks: Enhancing individual creativity through conflict and integration. Organizational Behavior and Human Decision Processes 116(2), 229–240 (2011)
24. Moe, N., Dingsoyr, T., Dyba, T.: A teamwork model for understanding an agile team: A case study of a scrum project. Information and Software Technology 52, 480–491 (2010)
25. Nonaka, I., Takeuchi, H.: The Knowledge Creating Company. Oxford University Press, USA (1995)
26. Rahim, A.: Managing Conflict in Organizations. Praeger (1992)
27. Ringstad, M.A., Dingsøyr, T., Brede Moe, N.: Agile process improvement: Diagnosis and planning to improve teamwork. In: O'Connor, R.V., Pries-Heje, J., Messnarz, R. (eds.) EuroSPI 2011. CCIS, vol. 172, pp. 167–178. Springer, Heidelberg (2011)
28. Robertson, J.: Requirements analysts must also be inventors. IEEE Software 22, 48–50 (2005)
29. Sutherland, J.: Agile can scale: Inventing and reinventing scrum in five companies. Cutter IT Journal 14, 5–11 (2001)
30. Takeuchi, H., Nonaka, I.: The new new product development game. Harvard Business Review (1986)
31. van Lamsweerde, A., Letier, E., Darimont, R.: Managing conflicts in goal-driven requirements engineering. IEEE Trans. Softw. Eng. 24(11), 908–926 (1998)
32. Zhang, X., Stafford, T.F., Dhaliwal, J.S., Gillenson, M.L., Moeller, G.: Sources of conflict between developers and testers in software development. Information & Management 51(1), 13–26 (2014)