The Relationship between Agile Methods and Open Source Software Development

Brian Fitzgerald, University of Limerick

Open Source Software (OSS) development and agile methods are in relatively close harmony in terms of a range of factors, including frequent releases of software, sub-division of work into smaller pieces, the use of non-traditional coordination mechanisms for software development, and a degree of self-assignment in work allocation (Abrahamsson et al. 2003). However, it is important to recognize that there are several points of potential friction between agile and OSS practices which could make their combination challenging.

Although a literature review in the area (Gandomani et al. 2012) did not find any empirical study of the integration of agile and OSS software development, a number of studies have considered the issue conceptually (e.g. Goldman and Gabriel 2005; Russo et al. 2009; Warsta and Abrahamsson 2003). These researchers have identified some potential points of friction between the two development approaches in terms of their execution. For example, Warsta and Abrahamsson (2003) concluded that the main points of difference between agile and OSS development are location of developers, size of development teams, the nature of the customer and the primary objective. Other researchers have drawn on the 12 principles of the Agile Manifesto to discuss the compatibility between agile and OSS (Goldman and Gabriel 2005; Russo et al. 2009). For example, there is disagreement on whether the espoused benefits of agile methods, such as close cooperation and frequent communication while maintaining simplicity, can be captured by OSS teams. Overall, the conclusion is that OSS can address most of the principles of the Agile Manifesto, although some points of friction do occur. Table 1 summarizes the areas in which agile and OSS appear be congruent and the areas in which there may be potential friction points.

Table 1: Comparison of Agile and OSS Practices

Factor	Agile Methods	OSS
Agile and OSS Congruence		
Frequent releases of software	• Strive toward working software at end of every sprint (e.g. Cockburn 2004; Stapleton 1997; Sutherland and Schwaber 2011)	• Emphasizes the maxim of '*release early, release often*' (Raymond 2001) • Time-based release management (Michlmayr et al. 2015)
Self-organizing teams	• Reflect the principle of empowered developer teams acting creatively to fulfil necessary roles and adapt to change. (Dyba 2000; Nerur et al. 2005; Stapleton 1997) • "It is people, not methodologies, who develop systems" (Fitzgerald 1998)	• Self-selection of developers to work on tasks within their expertise (Mockus et al. 2000) • Flat, peer-oriented and decentralized teams (Scacchi 2002)
Evolving nature of software requirements	• Requirements emergent and adapted based on customer interaction and feedback (Boehm 2002; Highsmith 2000)	• Developers provide a patch that provides new functionality and the requirement is "*asserted after the fact*" (Scacchi 2004).
Avoid heavy up-front planning	• Strive to eliminate "*heaviness*" of plan-driven methods (Erickson et al. 2005) and "*travel light*" (Beck 2000; Ambler 2002), tailoring when necessary (Fitzgerald et al 2000)	• No explicit system design or project plan (Boldyreff et al. 2003; Mockus et al. 2000)
Refactoring	• Continuous evolution of initial functionality and improvement through refactoring (e.g. Beck 2000)	• '*Maintenance as reinvention'* as an opportunity to improve code and design (Scacchi 2004)
Sub-division of work into smaller pieces	• Development of small increments with rapid cycles (Abrahamsson et al. 2003; Nerur et al. 2005)	• Code is built in small frequent increments (Sharma et al. 2002; Warsta and Abrahamsson 2003)
Attitude to Change	• Embrace change even late in development (Beck 2004; Williams and Cockburn 2003)	• Sophisticated version control systems to allow change to be incorporated and rolled back if necessary (Robbins 2005)
Potential Areas of Friction		
Location and mode of communication	• Preference for co-located development teams, facilitating face-to-face "*osmotic communication*" (Cockburn 2004) in daily stand-up meetings (e.g. Beck 2004; Schwaber and Beedle 2001)	• Distributed teams with developers working asynchronously and then 'lurking' on email and IRC mechanisms to learn about project status before starting new contributions (Tomayko 2012) • Source code as primary means of communication (Raymond 2001) as expressed by Torvalds (2000) "*Talk is cheap, show me the code*"
Size of development team	• Prefer small empowered teams (Dyba and Dingsoyr 2008; Stapleton 2007)	• Large development groups (Warsta and Abrahamsson 2003) with clearly distinct levels of empowerment (Crowston et al. 2012)
Simplicity	• Ensure simplest design possible – maximize the amount of work not done – "good enough" solution (Beck 2000)	• Redundancy as developers work in parallel on same functionality and best solution chosen – the principle of "*optimistic concurrency*" (Lakhani and von Hippel 2009)
Role of Customer	• Active involvement of customer sought with Scrum role of Product Owner to act as proxy (Schwaber and Beedle 2001)	• 'Developer is often the customer' in OSS (Koch and Schneider 2002) – "*scratching a personal itch*" (Raymond 2001)
Continuous Integration	• Core practice in Extreme Programming (Beck 2000)	• Difficult to achieve in OSS projects (Gruhn et al. 2013)

References

Abrahamsson, P., Warsta, J., Siponen, M.T., and Ronkainen, J. (2003) New Directions on Agile Methods: A Comparative Analysis, *25th International Conference on Software Engineering*, 2003, pp. 244 - 254.

Adolph, S., Kruchten, P., and Hall, W. (2012). Reconciling perspectives: A grounded theory of how people manage the process of software development. *Journal of Systems and Software*, *85*(6), 1269-1286.

Ågerfalk, PJ and Fitzgerald, B (2008) Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy, *MIS Quarterly*, Vol 32, No. 3, pp. 385-410

Amason, A., and Sapienza, H. (1997) The effects of top management team size and interaction norms on cognitive and affective conflict. *Journal of Management*, 23: 496-516.

Ambler S (2002) *Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*, John Wiley & Sons.

Andres, H. P. (2011) Shared mental model development during technlogy-mediated collaboration. *International Journal of e-Collaboration* 7(3), 14-30.

Beck K (2000) *Extreme Programming Explained: Embrace Change,* Addison-Wesley.

Beck, K (2004) *Extreme Programming Explained: Embrace Change (2nd Edition),* Addison-Wesley.

Beck, K and Boehm B (2003) Agility through discipline: a debate, *IEEE Computer*, June 2003, pp. 44-47.

Berdou, E. (2006) Insiders and outsiders: paid contributors and the dynamics of cooperation in community led F/OS projects. *Open Source Systems*. Springer US, 2006. 201-208.

Bird, C., Pattison, D., D'Souza, R., Filkov, V., and Devanbu, P. (2008) Latent social structure in open source projects. In *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering* (pp. 24-35). ACM.

Boehm B (2002) Get Ready for Agile Methods, with Care, *IEEE Computer* (35:1), pp. 64-69.

Boldyreff C, Lavery J, Nutter D and Rank S (2003) Open-Source Development Processes and Tools, *Proceedings of the 3rd Workshop on Open Source Software Engineering*, International Conference on Software Engineering Portland, Oregon.

Bowers, J., May, J., Melander, E., Baarman, M., and Ayoob, A. (2002). Tailoring XP for large system mission critical software development. In *Extreme Programming and Agile Methods—XP/Agile Universe 2002* (pp. 100-111). Springer Berlin Heidelberg.

Buchan, J. N., Pare, M., and Munhall, K. G. (2006). Spatial statistics of gaze fixations during dynamic face processing. *Social Neuroscience* 2(1) 1-13.

Cao, L., Mohan, K., Xu, P., and Ramesh, B. (2004). How extreme does extreme programming have to be? Adapting XP practices to large-scale projects. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference*.

Cao, L., Mohan, K., Xu, P., and Ramesh, B. (2009) A framework for adapting agile development methodologies. *European Journal of Information Systems* 18, 332–343

Capiluppi A, Stol K, and Boldyreff C (2012) Commercial Stakeholders in the Evolution of Open Source Software, in: Hammouda et al., *Proceedings of the 8th International IFIP WG2.13 Conference on Open Source Systems* (OSS), IFIP Advances in Information and Communication Technology (AICT), vol. 378, pp. 178-200.

Cataldo, M. and Herbsleb, J. D. (2013) Coordination Breakdowns and Their Impact on Development Productivity and Software Failures. *IEEE Transactions on Software Engineering*, 39(3), pp. 343-36.

Chiravuri, A., Nazareth, D., and Ramamurthy, K. (2011). Cognitive conflict and consensus generation in virtual teams during knowledge capture: Comparative effectiveness of techniques. *Journal of Management Information Systems, 28*(1), 311-350.

Cockburn, A (2004) *Crystal Clear: A Human-Powered Methodology for Small Teams*, Addison-Wesley.

Colazo J. A. and Fang, Y. (2010) Following the Sun: Temporal Dispersion and Performance in Open Source Software Project Teams. *Journal of the Association for Information Systems*: Vol. 11: Iss. 11, Article 4.

Corbin, J. and Strauss, A. (Eds.). (2008). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage.

Crowston, K. and Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, *18*(4), 65-85.

Crowston, K., Howison, J., and Annabi, H. (2006). Information systems success in free and open source software development: Theory and measures. *Software Process: Improvement and Practice*, *11*(2), 123-148.

Crowston, K., Wei, K., Howison, J., and Wiggins, A. 2012. Free/Libre open-source software development: what we know and what we do not know. *ACM Computing Surveys* 44, 2, Article 7 (February 2012).

Cummings, J, Espinosa, A and Pickering, C (2009) Crossing Spatial and Temporal Boundaries in Globally Distributed Projects: A Relational Model of Coordination Delay, *Information Systems Research*, 20(3), 420-439.

Dennis, A. R., Fuller, R. M., and Valacich, J. S. (2008) Media, tasks and communication processes: A theory of media synchronicity. *MIS Quarterly*, 32:3, 575-600.

Deshpande A and Riehle D (2008) Continuous integration in OSS development, in *Open Source Development: Communities and Quality*, Springer, US, pp.273-280.

Deutsch, M. (1969). Conflicts: Productive and destructive. *Journal of Social Issues*, *25*(1), 7-42.

Dyba, T (2000) Improvisation in Small Software Organizations, *IEEE Software*, no. 5, vol. 17, pp. 82-87.

Dyba T and Dingsoyr T (2008) Empirical studies of agile software development: a systematic review, *Information & Software Technology*, http://dx.doi.org/10.1016/j.infsof.2008.01.006

Eilhard, J. and Ménière, Y. (2009). A Look Inside the Forge: Developer Productivity and Spillovers in Open Source Projects. Available at SSRN 1316772.
Ellis, D. G. and Fisher, B. A. (1994). *Small group decision making: Communication and the group process*. 4th Ed, McGraw Hill, New York.
Erickson, J. Lyytinen, K and Siau, K (2005) Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research, *Journal of Database Management*, no. 4, vol. 16, pp. 88 – 100.
Espinosa, J. A., Slaughter, S. A., Kraut, R. E., and Herbsleb, J. D.(2007). Team knowledge and coordination in geographically distributed software development. *Journal of Management Information Systems*, 24(1),135-169.
Faegri, T, Dyba, T and Dingsoyr, T (2010) Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work, *Information & Software Technology*, Vol. 52, No. 10, 1118-1132.
Fang, Y. and Neufeld, D. (2009) Understanding Sustained Participation in Open Source Software Projects. *Journal of Management Information Systems*, 25:4, 9-50.
Feinberg, M., Willer, R., and Keltner, D. (2011). Flustered and Faithful: Embarrassment as a Signal of Prosociality. *Journal of Personality and Social Psychology* 1-17.
Figl, K. and Saunders, C. (2011) Team Climate and Media Choice in Virtual Teams. *AIS Transactions on Human-Computer Interaction* (3) 4, pp. 189-213
Fitzgerald, B. (1998). An Empirically-Grounded Framework for the Information Systems Development Process.
Fitzgerald, B. (2006) The Transformation of Open Source Software. *MIS Quarterly* 30(3) 587-598.
Fitzgerald, B., Hartnett, G. and Conboy K. (2006) Customizing Agile methods to software practices at Intel Shannon. *European Journal of Information Systems*. 15, 200-213.
Fitzgerald, B., Russo, N., & O'Kane, T. (2000). An empirical study of system development method tailoring in practice. ECIS 2000 Proceedings.
Fornell, C. and Larcker, D.F. (1981). Evaluating Structural Equation Models with Unobservable Variables and Measurement Error. *Journal of Marketing Research* 18(3):382-388.
Forrester (2013) Q3 2013 Global Agile Software Application Development Online Survey, (www.forrester.com)
Gandomani, T. J., Zulzalil, H., Ghani, A. A. A., and Sultan, A. B. M. (2012). A Systematic Literature Review on Relationship between Agile Methods and Open Source Software Development Methodology. *International Review on Computers & Software*, *7*(4).
Gefen, D and Straub, D. (2005). A Practical Guide to Factorial Validity Using PLS-graph: Tutorial and Annotated Example. *Communications of the Association for Information Systems* 16(5):91-109.
George, J. F., Carlson, J. R., and Valacich, J. S. (2013). Media selection as a strategic component of communication. *MIS Quarterly*, *37*(4), 1233-1251.
Goldman, R. and Gabriel, R. P. (2005). *Innovation happens elsewhere: Open source as business strategy*. Morgan Kaufmann.
Goth G (2007) Sprinting towards open source development, *IEEE Software*, Vol. 24, No. 1, pp.88-91.
Gruhn V, Hannebauer C, and John C. (2013) Security of public continuous integration services. In *WikiSym*, pages 15:1–15:10. ACM, 2013.
Guba, E. G. (1981). Criteria for assessing the trustworthiness of naturalistic inquiries. *ECTJ*, *29*(2), 75-91.
Hair, J.F., Black, W.C., Babin, B.J., and Anderson, R.E. (2009) *Multivariate Data Analysis* (7th ed.). Prentice Hall, Upper Saddle River , New Jersey.
Hauge, O., Ayala, C., and Conradi, R. (2010) Adoption of open source software in software-intensive organizations – a systematic literature review, *Information and Software Technology*, 52 (2010) 1133–1154.
Heerey, E. A. and Velani, H. (2006). Implicit learning of social predictions. *Journal of Experimental Social Psychology*, 46: 577-581.
Highsmith, J (2000) *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*, New York: Dorset House
Howison, J. and Crowston, K. (2014) Collaboration through open superposition: A Theory of the Open Source Way. *MIS Quarterly 38(1) pp. 29-50*.
Jarvenpaa, S. L. and Leidner, D. E. (1998). Communication and trust in global virtual teams. *Journal of Computer-Mediated Communication*, *3*(4).
Jehn, K. A. (1997). A qualitative analysis of conflict types and dimensions in organizational groups. *Administrative Science Quarterly*, 530-557.
Jehn, K. A. and Mannix, E. A. (2001) The Dynamic Nature of Conflict: A Longitudinal Study of Intragroup Conflict and Group Performance, *Academy of Management Journal* 44(2): pp. 238–51.
Koch, S. and Schneider, G. 2002. Effort, cooperation and coordination in an open source software project: GNOME. Infor. Syst. J. 12, 1, 27–42.
Korsgaard, M. A., Schweiger, D. M., and Sapienza, H. J. (1995). Building commitment, attachment, and trust in strategic decision-making teams: The role of procedural justice. *Academy of Management Journal*, *38*(1), 60-84.
Krishnamurthy, S., 2006. On the intrinsic and extrinsic motivation of open source developers. *Knowledge, Technology & Policy* 18 (4), 17–39.
Krishnamurthy, Sandeep; Ou, Shaosong; Tripathi, Arvind K. (2013). Acceptance of Monetary Rewards in Open Source Software Development, *Research Policy*, 43 (1).
Lakhani, K. and von Hippel E (2009) "No Managers Required: A case study of collaborative innovation using managerial toolkits" MIT Sloan School of Management Working Paper.
Lerner, J., and Schankerman, M. (2010). *The Comingled Code: Open Source and Economic Development*. Cambridge: MIT Press.

Lerner, J. and Tirole, J. (2002). Some simple economics of open source. *J. of Industrial Economics*, 50(2), 197-234.
Levy, M. and Germonprez, M. (2015). Is it Egalitarianism or Enterprise Strategy? Exploring a New Method of Innovation in Open Source. *Proceedings of the 21st Americas Conference on Information Systems, Puerto Rico*.
Lin, C., Standing, C., and Liu, Y. C. (2008). A model to develop effective virtual teams. *Decision Support Systems*, *45*(4), 1031-1045.
Lin, T.-C., Hsu, J.S.-C., Cheng, K.-T. and Wu, S. (2011) Understanding the role of behavioural integration in ISD teams: an extension of transactive memory systems concept. *Information Systems Journal*, 22, pp. 211-234.
Lindell, M. K. and Whitney, D. J. (2001). Accounting for common method variance in cross-sectional research designs. *Journal of applied psychology*,*86*(1), 114.
Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., and Kahkonen, T. (2004). Agile software development in large organizations. *Computer*, 37(12), 26-34.
Lyytinen K and Rose G.M. (2006) Information system development agility as organizational learning. *European Journal of Information Systems* 15(2), 183–199
Majchrzak, A., Rice, R. E., Malhotra, A., King, N., and Ba, S. (2000). Technology adaptation: The case of a computer-supported inter-organizational virtual team. *MIS Quarterly*, 569-600.
Mann, C and Maurer, F (2005) A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction, *Agile Development Conference*, 2005.
Maruping, L. M., Zhang, X., and Venkatesh, V. (2009). Role of collective ownership and coding standards in coordinating expertise in software project teams. *European Journal of Information Systems*, 18(4), 355-371.
McAllister, D. J. (1995). Affect-and cognition-based trust as foundations for interpersonal cooperation in organizations. *Academy of management journal*,*38*(1), 24-59.
Michlmayr M, Fitzgerald B, Stol K (2015) Why and How Should Open Source Projects Adopt Time-Based Releases?, *IEEE Software*. March/April 2015.
Miranda, S. M. and Bostrom, R. P. (1993). The impact of group support systems on group conflict and conflict management. *Journal of Management Information Systems*, 63-95.
Mockus, A, Fielding, R and Herbsleb, J, (2000) A Case Study of Open Source Software Development: The Apache Server, *22nd International Conference on Software Engineering, ICSE*, Limerick, Ireland.
Moe, N. and D. Šmite (2008) Understanding a Lack of Trust in Global Software Teams: A Multiple-Case Study. *Software Process Improvement and Practice* 13(3), pp. 217-231
Nerur, S. Mahapatra, R. and Mangalaraj, G. (2005) Challenges of migrating to agile methodologies. *Communications of the ACM* 72– 78.
Norris, J. S. (2004). Mission-critical development with open source software: Lessons learned. *Software, IEEE*, *21*(1), 42-49.
O'Connor, R.V. and S. Basri (2012) The Effect of Team Dynamics on Software Development Process Improvement. *International Journal of Human Capital and Information Technology Professionals*, 3(3), 13-26
Pearsall, M. J., Ellis, A. P., and Stein, J. H. (2009). Coping with challenge and hindrance stressors in teams: Behavioral, cognitive, and affective outcomes. *Org. Behavior and Human Decision Processes*, *109*(1), 18-28.
Pearson A. W., Ensley, M. D. and Amason, A. C. (2002) An assessment and refinement of Jehn's intragroup conflict scale. *International Journal of Conflict Management*, 13:2, 110 – 126.
Pedhazur, E. J. (1982). *Multiple regression in behavioral research: explanation and prediction* (2nd ed.). Fort Worth, TX: Harcourt Brace.
Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., and Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, *13*(3), 303-337.
Podsakoff, P., MacKenzie, S., Lee, J., and Podsakoff, N. (2003) Common Method Biases in Behavioral Research: A Critical Review of the Literature and Recommended Remedies. *Journal of Applied Psychology* 88 879-903
Qian, C., Cao, Q., and Takeuchi, R. (2013). Top management team functional diversity and organizational innovation in China: The moderating effects of environment. *Strategic Management Journal*, *34*(1), 110-120.
Raja, U., and Barry, E. (2005). Investigating quality in large-scale Open Source Software. In *ACM SIGSOFT Software Engineering Notes* 30 (4), pp. 1-4.
Ramesh, B., Cao, L., Mohan, K., and Xu, P. (2006). Can distributed software development be agile?. *Communications of the ACM*, *49*(10), 41-46.
Ravichandran, T. and Rai, A. (2000) Quality Management in Systems Development: An Organizational System Perspective. *MIS Quarterly* Vol. 24, No. 3 (Sep., 2000), pp. 381-415.
Raymond, E.S. (2001) *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly Media.
Richardson, H. A., Simmering, M. J., and Sturman, M. C. (2009). A tale of three perspectives: Examining post hoc statistical techniques for detection and correction of common method variance. *Organizational Research Methods*, 12(4), pp. 762-800.
Riehle, D., Riemer, P., Kolassa, C., and Schmidt, M. (2014). "Paid vs. Volunteer Work in Open Source." In *Proceedings of the 47th Hawaii International Conference on System Science* (HICSS 2014). IEEE Press, 2014.
Ringle, C. M., Wende, S., and Will A. (2005): SmartPLS 2.0 (beta), www.smartpls.de.
Robbins J (2005) Adopting Open Source Software Engineering (OSSE) Practices by Adopting OSSE Tools. In Perspectives on Free and Open Source Software, Joseph Feller, Brian Fitzgerald, Scott A. Hissam, and Karim R. Lakhani (Eds.). MIT Press, 245–264.
Rolandsson, B, Bergquist, M., and Ljungberg, J. (2011) Open Source in the Firm: Opening Up Professional Practices of Software Development, *Research Policy* (40:4), pp. 576-587.
Russo B, Scotto M, Sillitti A and Succi G (2009) *Agile Technologies in Open Source Development*. IGI Publishing.

Sawyer, S. and Guinan, P. J. (1998). Software development: Processes and performance. *IBM Systems Journal*, *37*(4), 552-569.

Sarker, S., Munson, C. L., Sarker, S., and Chakraborty, S. (2009). Assessing the relative contribution of the facets of agility to distributed systems development success: an Analytic Hierarchy Process approach. *European Journal of Information Systems*, 18(4), 285-299.

Sarker, S. and Sahay, S. (2003). Understanding virtual team development: An interpretive study. *Journal of the Association for Information Systems*, 4(1), 1.

Saunders, C. S. and Ahuja, M. K. (2006). Are all distributed teams the same? Differentiating between temporary and ongoing distributed teams. *Small Group Research*, *37*(6), 662-700.

Scacchi, W. (2002, February). Understanding the requirements for developing open source software systems. In *Software, IEE Proceedings* (Vol. 149, No. 1, pp. 24-39). IET.

Schaubroeck, J., Lam, S. S., and Peng, A. C. (2011). Cognition-based and affect-based trust as mediators of leader behavior influences on team performance. *Journal of Applied Psychology*, *96*(4), 863.

Schwaber K and Beedle, M (2001) *Agile Software Development with Scrum*. Upper Saddle River: Prentice Hall.

Sengupta, K. and Te'eni, D. (1993). Cognitive feedback in GDSS: improving control and convergence. *MIS Quarterly*, 87-113.

Shah, S. K. (2006) Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development, *Management Science* (52:7), pp. 1000-1014.

Sharma, S, Sugumaran, V and Rajagopalan B. (2002). A framework for creating hybrid open source software communities. *Information Systems Journal* 12, 1 (2002), 7–25.

Stapleton J (1997) *Dynamic Systems Development Method: The Method in Practice*, Addison-Wesley, UK.

Stotts, D., Williams, L., Nagappan, N., Baheti, P., Jen, D., and Jackson, A. (2003). Virtual teaming: Experiments and experiences with distributed pair programming. In *Extreme Programming and Agile Methods-XP/Agile Universe 2003* (pp. 129-141). Springer Berlin Heidelberg.

Strode, D. E. S. L. Huff, B. Hope and S. Link (2012) Coordination in co-located agile software development projects. *Journal of Systems and Software* (85) pp. 1222-1228.

Sulaman, S. M., Orucevic-Alagic, A., Borg, M., Wnuk, K., Host, M., and de la Vara, J. L. (2014, August). Development of Safety-Critical Software Systems Using Open Source Software--A Systematic Map. In *Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference* (pp. 17-24).

Sutherland, J. and Schwaber, K. (2011). The scrum papers: Nut, bolts, and origins of an Agile framework. *SCRUM Training Institute*

Tan, Y., Mookerjee, V., and Singh, P. (2007). Social capital, structural holes and team composition: Collaborative networks of the open source software community. *ICIS 2007 Proceedings*, 155.

Tiwana, A. (2004) An empirical study of the effect of knowledge integration on software development performance, *Information and Software Technology* 46 (13) pp. 899–906

Tomayko, R. (2012) Your team should work like an open source project [Online]. Available: http://tomayko.com/writings/adopt-an-open-source-processconstraints.

Torvalds, L (2000). https://lkml.org/lkml/2000/8/25/132 Linux Kernel Mailing List

Torvalds, L. and Diamond, D. (2001) *Just for Fun: The Story of an Accidental Revolutionary*. Harper Collins, NY.

van Wendel de Joode, R. (2004). Managing conflicts in open source communities. *Electronic Markets*, 14(2), 104-113.

Venkatesh, V., Brown, S. A., and Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quarterly*, *37*(1), 21-54.

Vidgen, R. and X. Wang (2009): "A Coevolving Systems Approach to the Organization of Agile Software Development". *Information Systems Research*, 20(3) 355-376.

Von Krogh, G., Haefliger, S., Spaeth, S., and Wallin, M. W. (2012). Carrots and rainbows: Motivation and social practice in open source software development. *MIS quarterly*, 36(2), 649-676.

Warsta, J. and Abrahamsson, P. (2003). Is open source software development essentially an agile method? In *Proceedings of the 3rd Workshop on Open Source Software Engineering*, pp. 143-147.

Webber, S. S. (2008). Development of cognitive and affective trust in teams a longitudinal study. *Small group research*, *39*(6), 746-769.

West, D. and Grant, T. (2010) Agile Development: Mainstream Adoption Has Changed Agility. *Forrester Research Inc*., Jan 20, 2010.

Williams L and Cockburn, A. (2003) Agile Software Development: It's about Feedback and Change, *IEEE Computer*, no. 6, vol. 36, pp. 39-43.

Wixom, B.H. and Todd, P.A. (2005). A Theoretical Integration of User Satisfaction and Technology Acceptance. *Information Systems Research* 16(1):85-102.

Wold, H. (1985). Systems analysis by partial least squares. In P. Nijkamp, L. Leitner, & N. Wrigley (Eds.), *Measuring the unmeasurable* (pp. 221-251). Dordrecht, the Netherlands: Marinus Nijhoff.