It's About Time: Applying Temporality to Software Development Teams

Mairead O' Connor
National University of Ireland,
Galway
Galway, Ireland
m.oconnor67@nuigalway.ie

OpenSym '17 Companion, August 23–25, 2017, Galway, Ireland
© 2017 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-5417-2/17/08.
https://doi.org/10.1145/3126673.3126682

Abstract

Most existing software development research adopts a very simplistic, 'clock-based' mechanistic interpretation of time and ignores the highly complex, multi-faceted, subtle and socially embedded nature of temporality. This is a significant limitation given that software development is a highly complex, socially embedded activity. This research applies temporality theory to examine software development teams. This research contributes to research and practice by (i) identifying any gaps, misconceptions or general conceptual issues in the application of temporal concepts to software development to date, (ii) examining the complexity of temporality that exist within software development teams (iii) examining the impact that such complexity may have, and (iv) identify strategies for resolving these temporality issues in software development. To achieve the objective of this study, the comprehensive temporality framework proposed by Ancona et al., (2001) is used to understand the various components of temporality within the context of software development.

Author Keywords

Information systems development; time; temporality; temporal dimensions; software development teams

ACM Classification Keywords

H.4.3 [**Information Systems Applications**]: *Time management (e.g., calendars, schedules)*

Introduction

Around the turn of the millennia, researchers begun focusing on the concept of time within an organisational context. There was even a surge of studies which applied temporal lenses to examine organisations (eg., Ancona *et al*., 2001, Sonnentag, 2012; Orlikowski and Yates, 2004). Researchers acknowledged and emphasized the importance of considering temporal constructs within an organisational context (eg., Bluedorn and Denhardt 1988; Saunders et al., 2004; Standifer and Bluedorn 2006).

At the same time, research focusing on software development teams became a popular research area (eg., Drury *et al*., 2012; Fitzgerald *et al.,* 2006; Stewart & Gosain, 2006; Ryan & O'Connor 2009). This rise in research addressed the ever-problematic environment in which these teams operate. Specifically, software development projects continually fail to meet time targets (Bartis and Mitev, 2008; Carlton, 2014; Keil *et al*., 2000; Whittaker, 1999).

The failure to conquer time targets is an obvious and continuous endemic in software development. Researchers promise better and faster contemporary and emergent software development methods (eg., Beck et al., 2001; Cusumano & Smith, 1995; Mahnič, 2013). However, studies which focus on time explicitly is ignored. Instead, time is studied merely implicitly and is rarely the main focus of the study.

Literature review

Considering the importance of time in software development and following numerous calls to research temporality within the context of software development teams and under dynamic conditions, it has not been explored explicitly or sufficiently in information systems research (Lee and Liebenau 2000; Saunders and Kim 2007; Waller et al., 2001). Within research, time is often ignored, narrowly conceived, poorly studied, or assumed to be linear, mechanistic and simple (Ancona et al., 2001). This was found in software development literature (Kavanagh and Araujo 1995; Nandhakumar 2002; Orlikowski and Yates 2002; Saunders and Kim 2007), the broader information systems (O' Riordan et al., 2013; Orlikowski and Yates 2002; Shen et al., 2014) and in organisational and management literature (Ancona et al., 2001; George and Jones 2000; Mitchell and James 2001; Roe 2008; Zaheer et al., 1999). However, no in-depth literature review of time in software development research has previously been published. Thus, we aim to sufficiently explore how the complexity of time is studied in software development.

In conducting the literature review, an efficient scientific method was chosen; a systematic literature review. A systematic literature review is a "means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest" (Kitchenham, 2004:1). This type of review offers a high quality (Dybå & Dingsøyr, 2008), transparent and replicable review (Leidner & Kayworth, 2006). This method offers the capability of summarising a large quantity of research publications (Fink, 2005), for studies which aim to address a clearly formulated question (Petticrew & Roberts, 2006).

This study will use the Okoli & Schabram (2010) systematic literature review guidelines to examine time in software development. We followed an eight-step guideline which is required for completion of a systematic literature review. Although any of these steps can be followed by researchers who complete a narrative literature review, following all steps are essential to conduct a scientifically rigorous systematic literature review (Okoli & Schabram, 2010). This guide is based on (Okoli & Schabram, 2010) but draws from similar best practices from three other systematic literature review guides; Kitchenham & Charters (2007), Levy & Ellis (2006) and Webster & Watson (2002).

Temporal framework

Temporality refers to an individual's experience of time (Caldas and Berterö 2012), which includes our relationship to time (Heidegger, 1927), and how we react to time (Fraisse, 1963). The concept of temporality has been applied in diverse areas: nursing (Caldas and Berterö 2012), community (Bastian, 2014) urban development (Wunderlich, 2013), sport (Allen-Collinson 2003), and information systems (Lee and Liebenau 2000). However, most studies conclude with advocating the importance of temporality and provide recommendations for further research but rarely contribute to practice.

Several temporal frameworks have emerged over the last four decades, mostly within organisational research (e.g. Ancona *et al*., 2001b; Bluedorn and Denhardt, 1988; Mosakowski and Earley, 2000; Orlikowski and Yates, 2002; Sonnentag, 2012). This study adopts the Ancona et al., (2001) framework for a number of reasons (i) it's the most comprehensive framework, (ii) it's distinguishable from other frameworks as it provides three categories which are interrelated, (iii) it provides a unique sub category which considers the temporal personality of an actor. Although the framework was developed in 2001, it is still the most holistic framework to date. While studies have looked at some of the temporal component of the framework (e.g. (Austin 2001); (Bluedorn et al. 1999); Gevers *et al.,* 2006; Waller *et al.,* 2001), none have looked at temporality as a broad concept or have contributed to adapting the framework. Shen *et al*., (2014, p. 3) advocates that this framework "synthesizes a large swath of temporal concepts across diverse areas of temporal studies and provides a common organising framework for these temporal constructs and variables". As many of the problems associated with software development are inherently people, social, and interpretive issues (Lyytinen and Rose 2006; Wastell and Newman 1993), this framework is pertinent to this study.

The Ancona et al., (2001) has three categories; conceptions of time, mapping activities to time and actors relating to time. *Conceptions of time* are the various types of time an organisation experiences (Ballard, 2008). Time can be conceptualised in many ways using many types of time (Shen *et al.,* 2014) such as linear time, clock time, uniform time, event time and cyclical time. The most popular and widely cited types of time are clock time and event time (Mosakowski and Earley, 2000). *Mapping activities to time* is about achieving a valid analysis of what happens over time during an activity or activities (Roe, 2008). This category is popular in research and explains how events and activities can be mapped to time. Mapping activities to time explains when an activity will begin, how long it may take and any fluctuations or patterns over the interval (Ancona et al., 2001). Insightful trends emerge when we consider how activities map to time. These trends can explain various temporal constructs such as an activities rate of completion, rhythms, jolts, interruptions and deadline behaviour. *Actors' relating to time refers to two concepts*. Firstly, temporal perception variables are used to reveal how actors perceive the continuum of time. Temporal personality refers to the temporal preferences of these actors and how they act in response to specific perceptions. The 'actors' in this instance may refer to an individual, team or organisation, and perception can be influenced by a multitude of factors. The temporal personality, for example, varies among the different cultures around the world (Mosakowski and Earley 2000). As Ancona *et al.,* (2001) alludes, it is important to examine time from the perspective of actors engaged in activities.

Contribution to industry

This research applies temporality theory to study software development teams. Flow based software development teams were examined, not just in relation to clock time, but broader temporal concepts such as event time, temporal perception, temporal personality, and synchronicity. It contributes to research and practice by (i) identifying which components of time are enabled by flow practices and how this is being achieved; (ii) identifying resulting challenges and implications of these; (iii) identifying potential strategies for resolving these challenges, and (v) developing a roadmap for future research on time in software development flow. It also provides an immediate practical contribution by identifying lessons drawn from the cases studied that may be applicable in future implementations of flow and indeed other software development techniques.

Data collection methods

This study will use a multiple-case design to allow a cross-case analysis to develop more sophisticated descriptions and powerful explanations (Benbasat et al., 1987; Miles and Huberman 1994). Leveraging the case study is valuable for "problems... in which research and theory are at their early, formative stages and sticky, practiced-based problems where the experiences of the actors are important and the context of action is critical" (Benbasat *et al*., 1987, p. 369). The rationale for adopting a case study strategy is to "capture the circumstances and conditions of an everyday or common place situation" (Yin, 2009, p. 48). Therefore, to fulfil the objective of this research, studying cases focuses on 'contemporary events' (Yin, 2009) and facilitates engagement with those 'living the case' (Stake, 2000) to unearth the perceptions of time in flow-based software development teams.

Case Selection and Background

Two cases of teams using software development flow techniques were studied. The cases were purposefully selected for a number of reasons. Firstly, the teams were applying flow techniques in a very complex, challenging environment where high profile clients with high demands were implementing the developed software in live transaction environments. Decision-making was in some cases highly time-pressurised with short time windows for identifying and correcting errors and significant repercussions if projects ran over time.

Secondly, all companies were actively using flow techniques in the teams studied (=> 2 months). Thirdly, there was continued direct access to interviewees involved in the flow projects selected, included on-going interviews, direct observations, participant observations and access to all project documents. This proved highly valuable when studying temporality, which, as stated earlier, is often subtle, nuanced, and is highly context-dependent, implicit and socially embedded.

Results

Based on categorises of the Ancona *et al.,* (2001) framework, the followed were identified; (i) the temporal characteristics of flow practices, examining the extent to which flow practices consider and effectively incorporate the various components of temporality, and (ii) the resulting temporal challenges of flow practices.

There were several dominant components of time that were present in the organisation. They were; (i) clock time, (ii) event time, (iii) subjective time, (iv) cyclical time, (v) socially constructed time, (vi) estimating, (vii) rate of completion, (viii) rhythm, (ix) interruptions, (x), deadline behaviour (xi) synchronization, (xii) temporal personality, and (xiii) time horizon.

There were three categories of problems; (i) problems associated with conceptions to time encountered by each team, (ii) problems associated with mapping activities to time encountered by each team, (iii) and problems associated with actors' relation to time encountered by each team.

References

1. Allen-Collinson, J. 2003. "Running into Injury Time: Distance Running and Temporality," Sociology of sport journal (20:4), pp. 331-370.
2. Ahmad, M.O., Markkula, J. and Oivo, M. 2013. Kanban in Software Development: A Systematic Literature Review. In 2013 39th Euromicro Conference on Software Engineering and Advanced Applications (pp. 9-16). IEEE.
3. Allen-Collinson, J. 2003. "Running into Injury Time: Distance Running and Temporality," Sociology of sport journal (20:4), pp. 331-370.
4. Ancona, D. G., Goodman, P. S., Lawrence, B. S., and Tushman, M. L. 2001a. "Time: A New Research Lens," Academy of management Review (26:4), pp. 645-663.
5. Ancona, D. G., Okhuysen, G. A., and Perlow, L. A. 2001b. "Taking Time to Integrate Temporal Research," Academy of Management Review (26:4), pp. 512-529.
6. Austin, R. D. 2001. "The Effects of Time Pressure on Quality in Software Development: An Agency Model," Information systems research (12:2), pp. 195-207.
7. Ballard, D.I., 2008. 11 Organizational temporality over time. *Time in organizational research*, *3*, p.204.
8. Bastian, M., (2014). Time and community: A scoping study. *Time & Society*, pp. 1-44.
9. Bartis.,E and Mitev., N. 2008. A multiple narrative approach to information systems failure: a successful system that failed. European Journal of Information Systems 17, 112–124.
10. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. and Kern, J., 2001. Manifesto for agile software development

11. Benbasat, I., Goldstein, D. K., and Mead, M. 1987. "The Case Research Strategy in Studies of Information Systems," Mis Quarterly (11:3), pp. 369-386.

12. Bluedorn, A. C., and Denhardt, R. B. 1988. "Time and Organizations," Journal of management (14:2), pp. 299-320.

13. Bluedorn, A. C., Kalliath, T. J., Strube, M. J., and Martin, G. D. 1999. "Polychronicity and the Inventory of Polychronic Values (Ipv) the Development of an Instrument to Measure a Fundamental Dimension of Organizational Culture," Journal of managerial psychology (14:3/4), pp. 205-231.

14. Boell, S.K. and Cecez-Kecmanovic, D., (2010). Literature reviews and the hermeneutic circle. Australian Academic & Research Libraries, 41(2), pp.129-144.

15. Caldas, C. P., and Berterö, C. 2012. "A Concept Analysis About Temporality and Its Applicability in Nursing Care," Nursing forum: Wiley Online Library, pp. 245-252.

16. Carlton, D. 2014. "Aaa-Rated Project Failures–Abdication, Avoidance and Apathy," Gartner Research document G (262544).

17. Clarke, P., O'Connor, R.V. and Leavy, B., 2016, May. A complexity theory viewpoint on the software development process and situational context. In Proceedings of the International Workshop on Software and Systems Process (pp. 86-90). ACM.

18. Conway, C. and Limayem, M., 2011. " You Want It When?" How Temporal Dissonance in IT Workers Contributes to Project Failures. In Proceedins of the Thirty Second International Conference on Information Systems, Shanghai 2011

19. Cusumano, M.A. and Smith, S.A., 1995. Beyond the waterfall: Software development at Microsoft. International Center for Research on the Management of Technology, Sloan School of Management, Massachusetts Institute of Technology.Vancouver

20. Dingsøyr, T., Nerur, S., Balijepally, V. and Moe, N.B. 2012. "A Decade of Agile Methodologies: Towards Explaining Agile Software Development," Journal of Systems and Software, (85:6), pp.1213-1221.

21. Drury, M., Conboy, K. and Power, K., 2012. Obstacles to decision making in Agile software development teams. Journal of Systems and Software, 85(6), pp.1239-1254.

22. Dybå, T. and Dingsøyr, T. 2008. "Empirical Studies of Agile Software Development: A Systematic Review," Information and software technology, (50:9), pp.833-859.

23. Fink, A., 2005. Conducting research literature reviews: From the internet to paper. Sage.

24. Fitzgerald, B., Hartnett, G. and Conboy, K., 2006. Customising agile methods to software practices at Intel Shannon. European Journal of Information Systems, 15(2), pp.200-213.

25. Fraisse, P. (1963). *The psychology of time*. New York: Harper.

26. Gevers, J.M., Rutte, C.G. and Van Eerde, W., 2006. Meeting deadlines in work groups: Implicit and explicit mechanisms. *Applied psychology*, *55*(1), pp.52-72.

27. George, J. M., and Jones, G. R. 2000. "The Role of Time in Theory and Theory Building," Journal of management (26:4), pp. 657-684.

28. Graziotin, D., Wang, X. and Abrahamsson, P., 2014. Happy software developers solve problems better: psychological measurements in empirical software engineering. PeerJ, 2, p.e289.

29. Heidegger, M. 1927. "Being and Time," *Oxford: Blackwell Publishing*).

30. Ikonen, M., Kettunen, P., Oza, N. and Abrahamsson, P. 2010. “Exploring the Sources of Waste in Kanban Software Development Projects”. In 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications (pp. 376-381). IEEE.
31. Kavanagh, D. and Araujo, L., 1995. Chronigami: Folding and unfolding time. Accounting, Management and Information Technologies, 5(2), pp.103-121.
32. Keil M, Mann J and Rai A.2000. Why software projects escalate: an empirical analysis and test of four theoretical models. MIS Quarterly 24, 631–664.
33. Khurum, M., Petersen, K. and Gorschek, T. 2014. “Extending Value Stream Mapping Through Waste Definition Beyond Customer Perspective,” Journal of Software: Evolution and Process, (26,12), pp.1074-1105.
34. Kitchenham BA, Pfleeger SL, Pickard LM, et al. (2002) Preliminary guidelines for empirical research in software engineering. IEEE Transactions on software engineering 28: 721-734.
35. Kitchenham, B., 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*(2004), pp.1-26.
36. Lee, H., and Liebenau, J. 2000. "Temporal Effects of Information Systems on Business Processes: Focusing on the Dimensions of Temporality," Accounting, Management and Information Technologies (10:3), pp. 157-185.
37. Leidner, D.E. and Kayworth, T., 2006. A review of culture in information systems research: Toward a theory of information technology culture conflict. MIS quarterly, 30(2), pp.357-399.
38. Levy, Y. and Ellis, T.J., 2006. A systems approach to conduct an effective literature review in support of information systems research. Informing Science, 9.
39. Lyytinen, K., and Rose, G. M. 2006. "Information System Development Agility as Organizational Learning," European Journal of Information Systems (15:2), pp. 183-199.
40. Mahnič, V., 2013. Applying Kanban Principles to Software Development. Information Technology for Practice 2013, p.89.
41. Miles, M. B., and Huberman, A. M. 1994. Qualitative Data Analysis: An Expanded Sourcebook. SAGE publications, Inc.
42. Mitchell, T. R., and James, L. R. 2001. "Building Better Theory: Time and the Specification of When Things Happen," Academy of Management Review (26:4), pp. 530-547.
43. Mosakowski, E. and Earley, P.C. 2000. “A Selective Review of Time Assumptions in Strategy Research,” Academy of Management Review, (25:4), pp.796-812.
44. Mühlhans, A. C. 2015. Time Concepts in Intercultural Business: India and Germany in Comparison. Munich: Grin-Verlag.
45. Nandhakumar, J. 2002. "Managing Time in a Software Factory: Temporal and Spatial Organization of Is Development Activities," The Information Society (18:4), pp. 251-262.
46. Okoli, C. and Schabram, K., 2010. A guide to conducting a systematic literature review of information systems research.
47. O Riordan, N., Conboy, K. and Acton, T. 2013. How soon is now? Theorising temporality in Information Systems research. In International Conference on Information Systems (ICIS) 2013: Reshaping Society Through Information Systems Design, Milan, Italy, 15-18 December 2013.
48. Orlikowski, W.J., and Yates, J.A. 2002. "It's About Time: Temporal Structuring in Organizations," Organization Science, (13:6), pp. 684-700.

49. Petticrew, M. and Roberts, H., 2006. How to appraise the studies: an introduction to assessing study quality. Systematic reviews in the social sciences: A practical guide, pp.125-163.

50. Power, K. and Conboy, K. 2015. A metric-based approach to managing architecture-related impediments in product development flow: an industry case study from Cisco. In Proceedings of the Second International Workshop on Software Architecture and Metrics (pp. 15-21). IEEE Press.

51. R. Pressman. 2005. Software Engineering a Practitioner's Approach. Boston, MA: McGraw-Hill,

52. Roe, R. A. 2008. "16 Perspectives on Time and the Chronometric Study of What Happens in Organizations," Time in organizational research (3), p. 291.

53. Ryan, S. and O'connor, R.V., 2009. Development of a team measure for tacit knowledge in software development teams. Journal of Systems and Software, 82(2), pp.229-240.

54. Saunders, C., and Kim, J. 2007. "Perspectives on Time," Management Information Systems Quarterly (31:4), p. 1.

55. Schwaber, K., 1997. Scrum development process. In Business object design and implementation (pp. 117-134). Springer London.

56. Shen, Z., Lyytinen, K., and Yoo, Y. 2014. "Time and Information Technology in Teams: A Review of Empirical Research and Future Research Directions," European Journal of Information Systems (24:5), pp. 492-518.

57. Sonnentag, S., 2012. Time in organizational research: Catching up on a long neglected topic in order to improve theory. Organizational Psychology Review, 2(4), pp.361-368.

58. Stake R. (2000) Case Studies' in Handbook of Qualitative Research, Denzin, N. and Y. *Sage, Thousand Oaks*.

59. Stewart, K.J. and Gosain, S., 2006. The impact of ideology on effectiveness in open source software development teams. Mis Quarterly, pp.291-314.
Vancouver

60. Waller, M. J., Conte, J. M., Gibson, C. B., and Carpenter, M. A. 2001. "The Effect of Individual Perceptions of Deadlines on Team Performance," Academy of Management Review (26:4), pp. 586-600.

61. Wang, X., Conboy, K. and Cawley, O. 2012. ""Leagile" Software Development: An Experience Report Analysis of the Application of Lean Approaches in Agile Software Development," Journal of Systems and Software, (85:6), pp.1287-1299.

62. Wastell, D., and Newman, M. 1993. "The Behavioral Dynamics of Information System Development: A Stress Perspective," Accounting, Management and Information Technologies (3:2), pp. 121-148.

63. Webster, J. and Watson, R.T., 2002. Analyzing the past to prepare for the future: Writing a literature review. MIS quarterly, pp.xiii-xxiii.

64. Whittaker B. 1999. What went wrong? Unsuccessful IT projects. Information Management and Computer Security 7, 23–29.

65. Wunderlich, F.M., 2013. Place-temporality and urban place-rhythms in urban analysis and design: An aesthetic akin to music. *Journal of Urban Design*, *18*(3), pp.383-408.

66. Yin, R. K. 2009. "Case Study Research: Design and Methods. 4. Udgave." Sage Publications.

67. Zaheer, S., Albert, S., and Zaheer, A. 1999. "Time Scales and Organizational Theory," Academy of Management Review (24:4), pp. 725-741.

68.