

Losing the Plot: Decision Behaviours in Agile Systems Development

Sharon Coyle
Business Information Systems Group
J.E. Cairnes School of Business & Economics
National University of Ireland Galway
Email: sharon.coyle@nuigalway.ie

Kieran Conboy
School of Information Systems, Technology & Management
University of New South Wales
Sydney 2052, Australia
Email: k.conboy@unsw.edu.au

Thomas Acton
Business Information Systems Group
J.E Cairnes School of Business & Economics
National University of Ireland Galway
Email: thomas.acton@nuigalway.ie

Abstract

The importance of effective decision making in organisations has been well documented. Groups are often formed in order to collaborate skills and information and assist with decision-making. Despite the many benefits associated with groupwork, groups are also subjected to process losses such as groupthink, which in turn have a significant impact on group decision-making. Such process losses may be even more prevalent in agile software development dominated by highly cohesive, self-managing teams. The purpose of this study is to investigate group decision-making in agile software development and assess how agile practices can reduce the occurrence of group process losses.

Keywords

Agile Systems Development, Process Loss, Agile Practices, Group Decision Making

INTRODUCTION

Research has shown that effective decision making is a critical component of organisational success where “high quality decisions are expected to lead to more productive actions, quicker problem solving and better organizational performance” (Eierman et al. 1995). The success of managers and leaders can hinge on the quality of their decision making, yet many organisations get decision making all wrong (Garvin et al. 2001). Due to the inherent complexity of decision making in organisations, groups are often formed so that individuals can share the information they have and generate new ideas (Mennecke 1997). Such group decision-making is recognition that ‘the whole is greater than the sum of its parts’ (Aristotle). Information Systems (IS) researchers and organisations have long recognised the need for general group-driven work (Janz 1999) because “the tacit nature of user requirements, project design specifications and overall project understanding cannot be fully captured in formal documents” (Janz et al. 2009). Unfortunately, research has shown how information exchange in group decision-making in general is often done poorly. In many instances, “discussion tends to focus on information that was already known and shared by the group prior to any interaction” (Hardman 2009). Such group process losses creates unease where systems development is concerned where key information is spread across a spectrum of stakeholders and the surfacing of such information for decision-making is a key necessity to the success of any Information Systems Development (ISD) project.

AGILE SYSTEMS DEVELOPMENT

One of the most recent and significant contending IS methodological approaches is that of agile software development. Agile development emerged as a result of continued pressure for “radical change in the traditional approach to development” whereby the “traditional life-cycle approaches that result in the eventual delivery of systems after several years” were no longer appropriate (Fitzgerald 1998). It was increasingly recognised that projects were “still over budget and behind schedule in far more cases than IS professionals and management” found acceptable (Kweku Ewusi-Mensan 1997). As a result, ISD saw further “suggestions for improvement” from “experienced practitioners who have labeled their methods *agile software development*” (Dyba et al. 2008). Agility (as it relates to ISD) can be defined as “iterative and evolutionary in development, planning and delivery to allow for rapid and flexible response to changes” (Batra et al. 2010). The Manifesto for Agile Software Development outlines a clear set of principles and beliefs underpinning agile methodologies (Batra et al. 2010; Williams et al. 2003) as follows:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

Agile places increasing emphasis on personal communication, community, morale, talent, skill and individual competency (Cockburn et al. 2001) and they “derive much of their agility by relying on the tacit knowledge embodied in the team, rather than writing the knowledge down in plans” (Boehm 2002). There are several agile methods utilised in practice and a detailed analysis of all methods is beyond the scope of this research. For the purpose of this research two of the most popular and widely adopted agile methodologies will be explored, which according to many researchers (e.g. Batra et al. 2010; Karlsson et al. 2000; Salo et al. 2008) are XP and Scrum. Scrum provides “an agile approach for managing software projects while increasing the probability of successful development of software, whereas XP focuses more on the project level activities of implementing software” (Salo et al. 2008). Both methodologies contain a detailed list of practices, which are presented in Tables 1 and 2.

Table 1. XP Practices

| Practices | Description |
|---------------------------|--|
| 1. Planning Game | Quickly determine the scope of the next release by combining business priorities and technical estimates. As reality overtakes the plan, update the plan. |
| 2. Small Releases | Put a simple system into production quickly, and then release new versions on a very short cycle. |
| 3. Metaphor | Guide all development with a simple shared story of how the whole system works |
| 4. Simple Design | The system should be designed as simply as possible at any given moment. Extra complexity is removed as soon as it is discovered |
| 5. Testing | Programmers continually write unit tests, which must be run flawlessly for development to continue. Customers write tests demonstrating that features are finished |
| 6. Refactoring | Programmers restructure the system, without changing its behaviour to remove duplication, improve communication, simplify or add flexibility |
| 7. Pair-Programming | All production code is written with two programmers at one machine |
| 8. Collective Ownership | Anyone can change any code anywhere in the system at any time |
| 9. Continuous Integration | Integrate and build the system many times a day, every time a task is completed |
| 10. 40-hour | Work no more than 40 hours a week as a rule. Never work overtime a |

| | | |
|-----|-------------------|--|
| | week | second week in a row. |
| 11. | On-site Customers | Include a real, live user on the team, available full-time to answer questions |
| 12. | Coding Standards | Programmers write all code in accordance with rules emphasising communication through the code |

Source: Beck (1999, pg. 47-48)

Table 2. Scrum Practices

| | Practices | Description |
|----|-------------------------|--|
| 1. | Scrum Master | Responsible for the success of Scrum by ensuring that the values, practices and rules are enacted and enforced. They are the driving force behind all the Scrum practices |
| 2. | Product Backlog | An evolving, prioritised queue of business and technical functionality that needs to be developed into a system |
| 3. | Scrum Teams | Commits to achieving a Sprint goal. They are accorded full authority to do whatever they decide is necessary to achieve the goal |
| 4. | Daily Scrum Meetings | Team comes to communicate daily for a 15-minute status meeting to determine what has been accomplished since the last meeting and what is going to be done before the next including any obstacles that are in the way |
| 5. | Sprint Planning Meeting | Customers, users, management, product owner and Scrum Team determine the next sprint goal and functionality and devises individual tasks that must be performed to build the product increment |
| 6. | Sprint | Team works for a fixed period of time |
| 7. | Sprint Review | Four-hour informational meeting. Team presents to management, customers, users and product owner the product increment that it has built during the Sprint |

Source: Schwaber et al. (2002, pg. 31-56)

Decision Making

According to Highsmith et al. (2001) “team proximity and intense interaction between team members are hallmarks of all agile methods.” A basic principle of agile is that “people can transfer ideas faster by talking face to face than by writing and reading documents. A few designers sitting together can produce a better design than each could produce alone” (Highsmith et al. 2001). Due the variation of stakeholders involved in any ISD project (plan or agile-driven), the project team will consist of a cohort of members with diverse interests, perspectives and skill-sets and as a result are undoubtedly subjected “to all the vagaries of group dynamics, interactions, coordination and communication” (Kweku Ewusi-Mensan 1997). This becomes particularly dominant in agile methodologies, which necessitate regular, intense stakeholder interaction and has a significant impact on group decision-making. In agile software development, work is always conducted by a self-managing team (Moe et al. 2010) who have “autonomy to make decisions that are traditionally the responsibilities of supervisors and managers” (Alper et al. 1998). Challenges associated with such devolved decision-making in an agile context become evident when combined with multiple diverse stakeholder perspectives and interests, daily intense group interaction and inevitable exposure to group process losses. Table 3 presents a list of process losses associated with groups.

Table 3. Group Process Losses

| | Description |
|----|---|
| 1. | Social pressures of conformity may result in <i>groupthink</i> (people begin to think alike and do not tolerate new ideas – yielding to conformance pressure) |
| 2. | Time-consuming, slow process (only one member can speak at a time) and high cost of meeting (travel, participation etc.) |
| 3. | Lack of coordination of the meeting and poor meeting planning |
| 4. | Inappropriate influences (dominance of time, topic, opinion by one or few individuals; fear of contributing because of the possibility of flaming and so on) |
| 5. | Tendency of group members to either dominate the agenda or rely on others to do most of the work (free-riding) |
| 6. | Some members may be afraid to speak up |
| 7. | Tendency to produce compromised solutions of poor quality |
| 8. | Non-productive time (socialising, preparing, waiting for latecomers – airtime fragmentation) |
| 9. | Tendency to repeat what was already said (because of failure to remember or process) |

Source: Turban et al. (2008)

Agile teams illustrate a strong potential for the occurrence of groupthink; one of the most commonly cited group process losses popularised by Janis (1971). Agile teams are inherently cohesive. A cohesive team is a “piori condition for agile approaches to work” (McAvoy et al. 2009). While group cohesion is positively associated with team performance, it is paradoxically a source of ineffective or dysfunctional decision making, the most notorious of which is groupthink (McAvoy et al. 2009). Groupthink occurs when “a team conforms to a strong leader’s opinion and has little tolerance for divergent opinions” (Janis 1971). Groupthink can lead a group to conform rather than search for an optimal decision during group decision making and such conformity pressure and conflict minimisation “result in less information sharing, poorer communication, fewer challenged assumptions and suboptimal decisions” (Lam 1997). This places an additional challenge to group decision-making in an agile software development context. Aldag et al. (1993) summarised Janis’ (1971) consequences of groupthink as follows:

- The group limits its discussion to only a few alternatives
- After a course of action is initially selected, members ignore new information concerning its risks and drawbacks
- The group avoids information concerning the benefits of rejected alternatives
- Group members make little attempt to use experts to obtain more precise information
- The group fails to consider what may go wrong and as such do not develop contingency plans

While research has indicated an intrinsic vulnerability of an agile team to groupthink (one of the group process losses presented in table 1), additional research is needed to assess the degree to which agile teams are exposed to other group process losses which impact decision making and how agile practices can be configured to reduce the occurrence and impact of such losses.

PROPOSED RESEARCH APPROACH

The study aims to assess how agile practices impact group decision-making and how they can be configured to reduce the occurrence and/or impact of group process losses associated with decision making. Due to the restrictions of a positivist research approach in its neglect of human behaviour and social factors (which are imperative in this study assessing group decision making in highly cohesive agile teams) an interpretative stance

will be followed. In addition, having considered the propositions pertaining to qualitative and quantitative research, a qualitative approach is considered most appropriate for this study, as there is little prior research on group decision-making in agile practices. There is a need to initially explore the field to extrapolate meaning for which a qualitative approach is best suited.

The intention is to conduct case study research of agile software development teams by conducting one-to-one interviews with project team members. In addition, observation of group decision making occurring within agile practices will be carried out. The case study, as described by Glatthorn et al. (2005, pg. 103) is a disciplined inquiry that develops an understanding of a particular subject matter through the use of inductive processes. Case studies are strongly associated with qualitative research as they "allow for the generation of multiple perspectives either through multiple data collection methods or through the creation of multiple accounts from a single method" which can yield detailed understanding of a specific context (Gray 2009, pg. 169). The case study will therefore derive meaning from events and develop knowledge in this research domain. In addition, as this research will occur in the natural setting of cases, there are opportunities for direct observation of team meetings and group decision making occurring within specific agile practices. It is anticipated that some relevant behavioural observations may give further insights (Yin 2009, pg. 109).

Current Status of the Project

Research to date has reviewed the literature on decision making and group decision making to include models of decision making, the decision making process, process losses associated with group decision making and decision making quality. In addition, literature on agile software development has been reviewed with a focus on agile practices associated with XP and Scrum. The research objective and research questions are defined. While case studies are the proposed research methodology, various research instruments utilised to assess group decision-making are currently being critiqued for their applicability to this study in an agile software development context. Once specified and pilot tested, data collection for this research will commence. Access has been granted in several organisations (some international) that are willing to participate in the research and are already using agile methodologies.

REFERENCES

- Aldag, R.J., and Fuller, S.R. "Beyond Fiasco - A Reappraisal of the Groupthink Phenomenon and a New Model of Group Decision-Processes," *Psychological Bulletin* (113:3), May 1993, pp 533-552.
- Alper, S., Tjosvold, D., and Law, K.S. "Interdependence and Controversy in Group Decision Making: Antecedents to Effective Self-Managing Teams," *Organizational Behavior and Human Decision Processes* (74:1), Apr 1998, pp 33-52.
- Batra, D., Xia, W., D., V., and Dutta, K. "Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry," *Communications of the Association for Information Systems* (27:21) 2010.
- Beck, K. *Extreme Programming Explained* Addison-Wesley, Reading, MA., 1999.
- Boehm, B. "Get Ready for Agile Methods, With Care," *Computer* (35:1), Jan 2002, pp 64-+.
- Cockburn, A., and Highsmith, J. "Agile Software Development: The People Factor," *Computer* (34:11), Nov 2001, pp 131-133.
- Dyba, T., and Dingsoyr, T. "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology* (50:9-10) 2008, pp 833-859.
- Eierman, M.A., Niederman, F., and Adams, C. "DSS Theory: A Model of Constructs and Relationships," *Decision Support Systems* (14:1), May 1995, pp 1-26.
- Fitzgerald, B. "An Empirical Investigation into the Adoption of Systems Development Methodologies," *Information & Management* (34:6) 1998, pp 317-328.
- Garvin, D.A., and Roberto, M.A. "What You Don't Know About Making Decisions," *Harvard Business Review* (79:8), Sep 2001, pp 108-+.
- Glatthorn, A., and Joyner, R. *Writing a Winning Thesis or Dissertation: A Step-by-Step Guide*, (Second ed.) Sage Publications Limited, 2005.
- Gray, D.E. *Doing Research in the Real World*, (2nd ed.) SAGE Publications Ltd., 2009.
- Hardman, D. *Judgement and Decision Making: Psychological Perspectives* BPS Blackwell, 2009.

- Highsmith, J., and Cockburn, A. "Agile Software Development: The Business of Innovation," *Computer* (34:9) 2001, pp 120-127.
- Janis, I.L. "Groupthink," *Psychology Today* (5:6) 1971.
- Janz, B.D. "Self-Directed Teams in IS: Correlates For Improved Systems Development Work Outcomes," *Information & Management* (35:3) 1999, pp 171-192.
- Janz, B.D., and Prasarnphanich, P. "Freedom to Cooperate: Gaining Clarity Into Knowledge Integration in Information Systems Development Teams," *IEEE Transactions on Engineering Management* (56:4), Nov 2009, pp 621-635.
- Karlsson, E.-A., Andersson, L.-G., and Leion, P. "Daily Build and Feature Development in Large Distributed Projects," in: *Proceedings of the 22nd international conference on Software engineering*, ACM, Limerick, Ireland, 2000.
- Kweku Ewusi-Mensan, K.C. "Critical Issues in Abandoned Information Systems Developments Projects," *Communications of the ACM* (40:9) 1997, pp 74-80.
- Lam, S.S.K. "The Effects of Group Decision Support Systems and Task Structures on Group Communication and Decision Quality," *Journal of Management Information Systems* (13:4), Spring97 1997, pp 193-215.
- McAvoy, J., and Butler, T. "The Role of Project Management in Ineffective Decision Making within Agile Software Development Projects," *European Journal of Information Systems* (18:4), Aug 2009, pp 372-383.
- Mennecke, B.E. "Using Group Support Systems to discover hidden profiles: An Examination of the influence of Group Size and Meeting Structures on Information Sharing and Decision Quality," *International Journal of Human-Computer Studies* (47:3), Sep 1997, pp 387-405.
- Moe, N.B., Dingsøy, T., and Dybå, T. "A Teamwork Model for Understanding an Agile Team: A Case Study of a Scrum Project," *Information and Software Technology* (52:5) 2010, pp 480-491.
- Salo, O., and Abrahamsson, P. "Agile Methods in European Embedded Software Development Organisations: A Survey on the Actual Use and Usefulness of Extreme Programming and Scrum," *IET Software* (2:1) 2008, pp 58-64.
- Schwaber, K., and Beedle, A. *Agile Software Development with SCRUM* Prentice Hall, Upper Saddle River, NJ, 2002.
- Turban, E., Leidner, D., McLean, E., and Wetherbe, J. *Information Technology for Management: Transforming Organizations in the Digital Economy* John Wiley & Sons, Inc., 2008.
- Williams, L., and Cockburn, A. "Agile software development: It's about feedback and change," *Computer* 2003, pp 39-43.
- Yin, R.K. *Case Study Research. Design and Methods*, (4th ed.) SAGE Publications, Inc., 2009.

COPYRIGHT

© 2010. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.