# EMERGENT DECISION-MAKING PRACTICES IN TECHNOLOGY-SUPPORTED SELF-ORGANIZING DISTRIBUTED TEAMS

*Research-in-progress submission*
*General Track*

**Robert Heckman**
Syracuse University
Syracuse, NY U.S.A.
rheckman@syr.edu

**Kevin Crowston**
Syracuse University
Syracuse, NY U.S.A.
crowston@syr.edu

**Qing Li**
Syracuse University
Syracuse, NY U.S.A.
qli03@syr.edu

**Eileen Allen**
Syracuse University
Syracuse, NY U.S.A.
eeallen@syr.edu

**U. Yeliz Eseryel**
Syracuse University
Syracuse, NY U.S.A.
uyeserye@syr.edu

**James Howison**
Syracuse University
Syracuse, NY U.S.A.
jhowison@syr.edu

**Kangning Wei**
Syracuse University
Syracuse, NY U.S.A.
kwei@syr.edu

## Abstract

*We seek to identify work practices that make technology-supported self-organizing distributed (or virtual) teams (TSSODT for short) effective in producing outputs satisfactory to their sponsors, meeting the needs of their members and continuing to function. A particularly important practice for team effectiveness is decision making: are the right decisions made at the right time to get the work done in a way that satisfies team sponsors, keeps contributors happy and engaged, and enables continued team success? In this research-in-progress paper, we report on an inductive qualitative analysis of 120 decision episodes taken by 2 Free/libre Open Source Software development teams (the completed paper will include 360 episodes from 6 teams). Our analysis revealed differences in the performance of the two teams that seems to be related to differences in overall project effectiveness.*

**Keywords:** Decision making practices, technology-supported teams, distributed teams, self-organizing teams, virtual teams, free/libre open source software development teams, team effectiveness, leadership

# Introduction

We seek to identify work practices that make technology-supported self-organizing distributed teams (TSSODT for short) effective in producing outputs, meeting member needs and continuing to function. A particularly important practice for team effectiveness is decision making: are the right decisions made at the right time to get the work done in a way that satisfies team sponsors, keeps contributors happy and engaged, and enables continued team success? We examine this question in the context of TSSODT because the decision-making practices of these teams emerge from the interactions of the team members rather than from organizational context. However, discontinuities between team members make such emergence and indeed any kind of decision process harder to attain. Since TSSODT are increasingly used in a variety of settings, it is important that we understand how to make them effective. As well, examination of decision-making practices can illuminate other issues critical to team effectiveness (e.g., team leadership), as we will discuss further below.

The contribution of this work-in-progress paper is to describe the range and evolution of decision-making practices in TSSODT based on a preliminary analysis of 120 decision episodes carried out by 2 naturally occurring teams. (For the final paper, we plan to include 360 episodes from 6 teams and to link decision-making practices to team factors, patterns of leadership and measures of overall team effectiveness.) In the remainder of this section, we briefly introduce the setting for our study before turning to a brief review of the relevant literature. Subsequent sections of the paper will describe our study methodology and preliminary findings. We will conclude with some discussion of our early results and plans for completing the research in progress.

Our study of decision-making practices is set in the context of TSSODT. Technology-supported distributed (or virtual) teams are of great interest to organizations because of their ability to bridge discontinuities of time and geography to enable access to and transfer of knowledge across boundaries thus leveraging human and intellectual capital (Duarte et al. 2001). Because they can quickly bring together the specific expertise needed to solve immediate problems regardless of geographical location, virtual teams permit organizations to respond quickly to unexpected changes in the environment and to non-routine problems. As a result, virtual teams form an increasingly important part of an organization's adaptive capability to respond to uncertainty and complexity and thus are prevalent.

Virtual teams also provide a mechanism for organizations to integrate diverse forms of specialized knowledge (Grant 1996). However, teams of diverse expert knowledge workers are hard to manage in conventional hierarchical terms. Members of these teams may come from a variety of organizations; rather than being assigned to the team by a common manager, members often voluntarily choose to participate. As a result, these teams are often self-organizing. Examples of self-organizing teams include *ad hoc task groups* that quickly form and dissolve, *voluntary learning groups* (e.g. *communities of practice*, *action learning groups*, or *study circles*) that may be informal or semi-formal or *self-managing work groups* within formal organizations. As organizations become increasingly knowledge-based and dependent on effective coordination of specialized knowledge for competitive advantage, self-organizing teams grow in importance. These two trends, towards increased virtuality and increased self-organization, lead to our interest in TSSODT.

In this paper, we examine in particular the decision-making practices of Free/Libre and Open Source Software (FLOSS)[1] development project teams. Core FLOSS developers comprise a work team because they have a shared goal of developing and maintaining a software product, are interdependent in terms of tasks and roles, and have a user base to satisfy, in addition to having to attract and maintain members. Although we focus on FLOSS teams, our findings will have implications for understanding all kinds of TSSODT because FLOSS development is an example of self-organized distributed work. Developers contribute from around the world, meet face-to-face infrequently (some not at all), and coordinate their activity primarily by means of ICT (Raymond 1998b; Wayner 2000). Team members typically contribute to projects without being employed by a common organization. While many contribute without being paid at all, it is increasingly common for companies to pay developers to work on FLOSS projects, giving these projects many of the characteristics of inter-organizational alliances (Fitzgerald 2006). Nevertheless, the teams are largely self-organizing, most often without formally appointed leaders or indications of rank or role. As a result, these teams depend on processes that span traditional boundaries of place and ownership (Watson-

---

[1] The free software movement and the open source movement are distinct and have different philosophies but mostly common practices. This paper focuses on team development practices, which we expect to be largely shared across the distributed teams in both movements. However, in recognition of these two communities, we use the acronym FLOSS, standing for Free/Libre and Open Source Software, rather than the more common OSS.

Manheim et al. 2002). Because many FLOSS teams have been quite effective in developing high quality software, study of these teams may reveal practices of widespread interest.

## Literature review

In this section, we very briefly review literature relevant to our study of decision-making practices in TSSODT. We consider as a team decision one that binds the team as a whole to a particular course of action. Many kinds of decisions have to be made collectively by a FLOSS development team, ranging from procedural questions such as which patches to accept, when to release a new version and with what content, (e.g., Erenkrantz 2003) to strategic questions such as overall direction for development or system architectures, to group maintenance questions, such as whom to accept as a developer. We've identified these as team decisions since they affect and bind the entire group and the software.

Of course, there is a huge literature on decision making in teams that is potentially relevant to our research but to which it is impossible to do justice within the space limitations of a research-in-progress paper. Broadly speaking though, these studies underscore the close tie between effective decision making and overall team effectiveness and the importance of understanding the practices by which decisions are actually made in teams. In the information systems literature more particularly, there have been numerous studies of ICT support for group decision making (e.g., DeSanctis et al. 1987; Fjermestad et al. 1998/1999; Turoff et al. 1993). Many of these studies have been design focused, offering important suggestions for systems to improve the process and quality of team decisions. For example, many studies have examined the value of structuring decision-making processes using software tools. Studies of groups in action have tended to adopt experimental methods and to focus on single episodes of decision making rather than on practices over life of an intact team (though there are exceptions, such as (Eden et al. 2001)). However, broadly speaking, there is a relative lack of studies that examine what kinds of decision processes emerge in intact self-organizing teams, how these practices evolve over time, and how they contribute to overall team effectiveness.

Examination of decision-making practices also sheds light on other team practices, in particular, on team leadership. Indeed, the two concepts—leadership and decision-making—are often equated, as in President Bush's recent pronouncement, "I'm the decider". On the other hand, Peter Drucker writes: "The least effective decision makers are the one who constantly make decisions. The effective ones make very few" (Drucker 1990), suggesting a more nuanced relationship. Leadership is of particular interest because leadership in self-organizing distributed teams is often emergent rather than appointed. In the absence of formally designated leaders, members within the team lead on a "voluntary" basis, either individually or collectively. According to Berdahl (1996), leaders emerge when "one or more of a group composed initially of equal status peers… exhibits notably higher levels of leadership behaviour and thereby attains higher status in the eyes of fellow group members" (p. 26). Some teams will evolve a leadership structure in which a single member emerges who is recognized by other members as the team's leader, while other teams will evolve a less-centralized leadership structure based on interaction and influence patterns. In the latter case, leadership can be shared among two or more team members or distributed among all team members (House et al. 1997; Pearce et al. 2003). In these groups a very different form of leadership seems to be at work. No single individual plays an obviously dominant role. When asked who their leaders are, members of these groups will often say, "We have no leaders". Still, this outcome is consistent with the functional approach to leadership. According to this view, some behaviours are viewed serving as leadership functions in that they help the team to achieve its goals and perform effectively. Through the interactions of the team members, one or more individuals emerge to perform the leadership behaviours that the team requires. More than one individual may perform leadership behaviours, and different individuals may perform the same leadership behaviours at different times (Pavitt 1998).

Decision making and leadership have been examined in several studies of FLOSS teams. Similar to studies of Group Decision Support Systems, some authors have noted the possibility of using software to encode and enforce particular work practices (Halloran et al. 2002). Case studies of FLOSS teams have documented a wide range of decision-making styles. A common concern is participation in decision making. At one extreme is a style where decisions are taken primarily by a few central participants, even a single individual, as in Linux, where Linus Torvalds originally made most of the decisions for the team (Moon et al. 2000). Such a decision style has been characterized as a "benevolent dictatorship" (Raymond 1998a). On the other extreme are teams with a decentralized communications structure and more consultative decision-making style. Some teams even settle decisions by voting. Of course, the decision style might be different for different kinds of decisions (e.g., decentralized for patches but centralized for strategic decisions). As well, the style might evolve over time as the project evolves. Fitzgerald

(2006) suggests that a small group will control decision making early in the life of a project, but as the project grows, more developers will be involved and German (2003) documents such a transition in the case of the Gnome project. What is not yet clear is which of these styles is most effective and in what situations. Our paper attempts to address this question.

## Methods and Data

To analyze the decision-making norms and processes of open-source projects, our research employs a comparative case study methodology, depending primarily on content analysis of the decision-making discussions. Because the team members interact primarily via ICT, to find these discussion, we analyzed the email discourse between administrators, developers, and users that takes place on the developers' e-mail lists. By examining on decision-making, we focus our attention on a particularly meaningful subset of the total collection of messages.

### Case site selection

We picked six FLOSS projects to examine in detail. Characteristics of the projects are given in Table 1. In order to maintain a balance between the maximization of variability and control of unwanted systematic variance, we selected six projects by considering several dimensions. First, we controlled for topic. We picked three projects that develop Instant Messenger (IM) software products (Gaim, Fire, and aMSN) and three that develop Enterprise Resource Planning (ERP) software products (Compiere, WebERP, OFBiz). We chose to examine projects on two topics to provide some comparison across very different kinds of projects These two types of projects are fundamentally different in the scope of the user universe they represent: IM clients are more or less contained within the world of a server protocol and represent the world of a single user, while enterprise software may be spread across multiple servers and represents an entire organization and its transactions. On the other hand, within these topics, the projects are essentially competitors, making comparisons between these projects valid.

Second, we picked projects that were roughly similar in age and status. All of the projects selected have released software and all describe their current status as production/stable. Five of the six projects are hosted on SourceForge, providing some control for potential differences in development tools (which as noted, could be used to structure decision making).

Finally, we looked for projects that had some variance in terms of effectiveness. We consider effectiveness as a multi-dimensional construct, including success of the project's outputs, team member satisfaction and continued project activities (Hackman 1987). We therefore applied the multivariate approach to success suggested by Crowston et al. (In press) and looked at downloads as an indication of project output, as well as the number of developers and users attracted to the product over time as indications of member satisfaction and continued team performance. In generally, number of downloads is too noisy a measure to be useful, but given the controls we applied in selecting projects, we feel it will provide information about the comparative adoption of projects of a single topic.

**Table 1. Characteristics of projects selected for analysis.**

|  | **Gaim** | **Fire** | **aMSN** | **Compiere** | **OFBiz** | **WebERP** |
|---|---|---|---|---|---|---|
| **Product** | Instant Messenger Client | Instant Messenger Client | Instant Messenger Client | Enterprise-level software | Enterprise-level software | Enterprise-level software |
| **Intended Audience** | End-user | End-user | End-user | Industry | Industry | Industry |
| **Programming language** | C | C | C | Java | Java | PHP |
| **License** | GNU General Public License (GPL) | GNU General Public License (GPL) | GNU General Public License (GPL) | Mozilla Public License 1.1 (MPL) | MIT Open Source License | GNU General Public License (GPL) |

| Developer count | 17 | 12 | 19 | 61 | 31 | 6 |
|---|---|---|---|---|---|---|
| Development site | SourceForge | SourceForge | SourceForge | SourceForge | ofbiz.org | SourceForge |

### *Unit of Analysis: Decision Episodes*

To analyze the email discussions, we selected decision episodes as our primary unit of analysis. We chose episodes rather than words, sentences, paragraphs, messages, or thematic units because we believe that the decision process we wish to understand is at its essence a process of interaction. In FLOSS projects, decision making occurs as a process that unfolds when project administrators, developers, and users interactively respond to triggers which present opportunities for the group to choose. Thus, we looked for sequences of messages that began with opportunities for the group to choose, and culminated in an observable or inferable choice. We defined a decision episode as a group of email messages that reflect as complete picture as possible of the process of decision making related to an issue. The decision episode begins with a message containing a trigger that presents an opportunity for choice (such as a feature request, a software bug report or a presentation of a strategic problem). It includes discussion related to the issue, and a decision concerning the stated opportunity.

In order to observe potential changes in decision-making processes and norms over time, we sampled 20 decision episodes from three time periods for each project. The Beginning period for each project consisted of the first 20 decision episodes observable on the developer mailing list. The Ending period for each project consisted of the last 20 decision episodes observable on the developer mailing list. The Middle period for each project consisted of 20 episodes surrounding a major code release approximately halfway between the Beginning and Ending periods. Figure 1 shows the time periods sampled for each project.

**Figure 1. Timing of early, middle and end sampling periods per project**
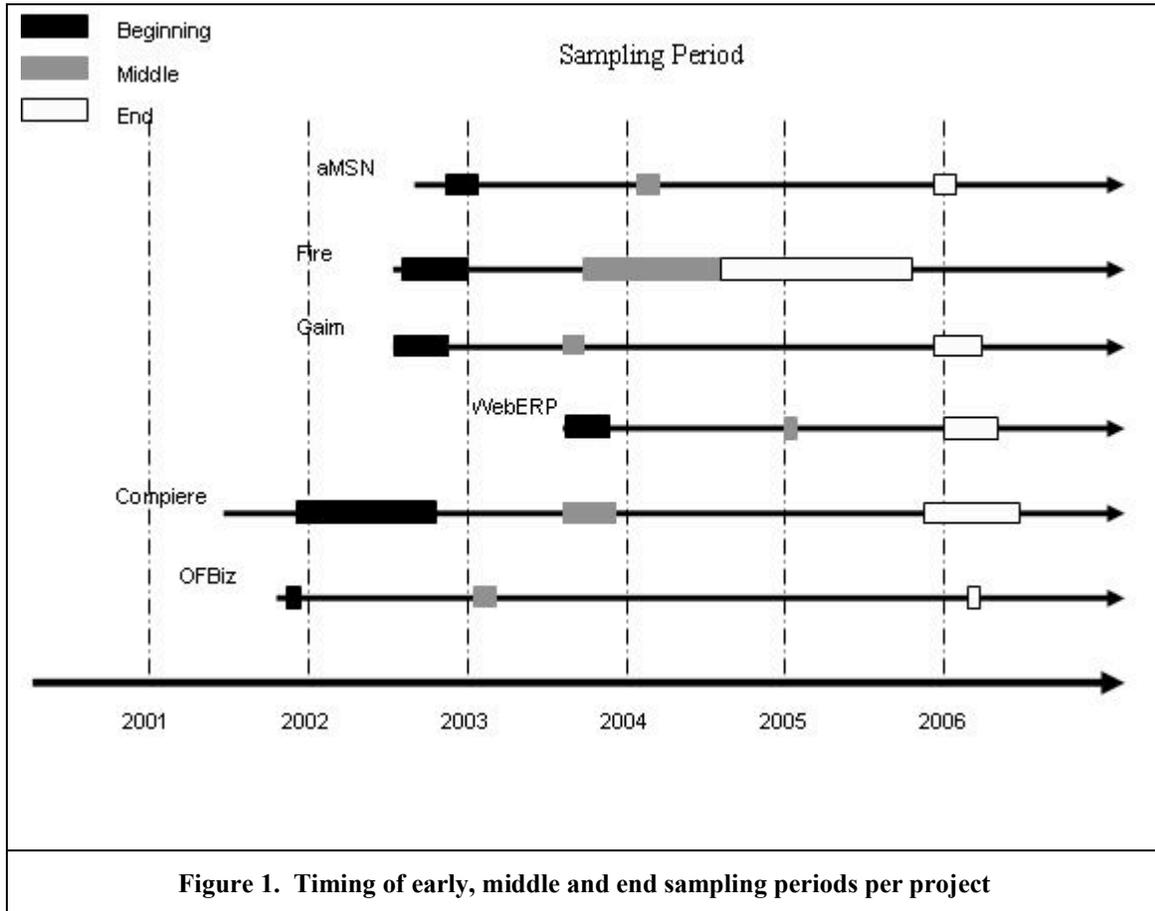
### Analysis and Coding of Episodes

We began analysis of decision episodes by coding observable, manifest elements of content that would provide consistent descriptions of episodes. We coded the number of messages per episode, duration of the episode (in days), total number of participants in the episode (including number of project administrators, number of developers, and number of users), and the role (administrator, developer, or user) of each message's sender.

Subsequent coding was inductive, with three independent analysts reading the episodes in order to understand the salient features of the decision process. Through several iterations, these independent coders identified and agreed upon five additional latent variables that were important to the decision process. These are decision type, decision trigger type, decision process complexity, decision announcement, and decision style.

***Decision Type***. Initial inspection of the discourse in developer mailing lists suggested that two types of choice opportunities might be fruitful to systematically observe: CODE decisions, and Non-CODE decisions. The first type of episode are those that involve decisions about the code itself – about bug fixes, additions of new features, or enhancement of the final product through a change in code. The second type (Non-CODE) were those that involve other kinds of decisions such as choices about strategic direction, infrastructure, alliances and partnerships, etc. They do not result in changed code.

***Decision Trigger Type.*** One goal of our inductive content analysis was to understand the types of triggers that presented opportunities for the group to choose. Thus, our coding has developed a preliminary and partial typology

of triggers. Decision episodes about CODE are triggered by: (1) bug reports, (2) feature requests, (3) problem reports, (4) patch submissions (5) release to-do lists and (6) mixed (bug and feature) lists not specifically associated with releases. We have not yet systematically identified the trigger types for Non-CODE episodes, and have thus far coded them as "Other."

***Decision Process Complexity***. Inductive analysis also indicated that some episodes required more complex decision choices than others. For example, some episodes involve a single choice that responds to a single straightforward trigger. These were coded as "Single." Others responded in a linear, straightforward fashion to a trigger that contained multiple opportunities for choice (e.g. a release to-do list.) These were coded as "Multiple-Simple." The most complex episodes were not straightforward or linear in nature. Regardless of the nature of the initial trigger, these episode developed new, sometimes unrelated, opportunities for choice throughout their life. The initial problem(s) might be solved or remain unsolved, and the new problems introduced might also remain unsolved. These episodes, which closely resemble the garbage can decision opportunities described by Cohen et al ( ), were coded "Multiple-Complex."

***Decision Announcement.*** In order to reliably determine that a decision had truly been reached, our independent coders coded the statement(s) that confirmed that a decision had been reached.

***Decision Style***. We are in the process of developing a typology of decision styles. The decision style will be coded for each episode. We will thus be able to determine if a project uses a single style, multiple styles, a predominant style, etc. We are currently developing a typology of decision styles.

To date we have reliably coded episodes from two IM projects (Fire and aMSN) on all variables except Decision Style. Episodes have been independently coded by two analysts, and their inter-rater agreement is over 90% for each variable.

## Preliminary Findings

We have completed preliminary analysis of 120 episodes from two projects. In this section, we present several elements of that analysis.

### *Duration, Length, and Participation in Decision Episodes.*

Table 2 shows how decision episodes evolved in terms of duration and length. In Fire, the average decision episode lasted for 3.5 days in the early sampling period. Duration steadily grew until the average episode lasted 5.2 days during the final sampling interval. The number of messages, however, in a Fire decision episode decreased from 7.6 to 5.1 over the same time frame.

In MSN, duration and length evolved differently. The average decision episode lasted only 2.6 days in the earliest sampling period. Duration rose slightly in the middle period, then fell to 2.2 days in the final period. Unlike fire, the number of messages in a decision episode steadily increased over time, increasing to 16 messages per episode by the final sampling period.

| Table 2 Average Duration & Number of Message per Episode | | | | | | |
|---|---|---|---|---|---|---|
| | **Beginning** | | **Middle** | | **End** | |
| | **Duration** | **# Message** | **Duration** | **# Message** | **Duration** | **# Message** |
| **Fire** | 3.55 | 7.6 | 4.3 | 5.55 | 5.15 | 5.1 |
| **aMSN** | 2.6 | 4 | 2.8 | 6.2 | 2.2 | 16.45 |

A one-way ANOVA showed that differences were not statistically significant in Fire. (Duration: $F=.58$, $df=2$; $p=.57$; Number of messages: $F=2.15$; $df=2$; $p=.13$). Though duration in aMSN is relatively consistent ($F=.16$, $df=2$; $p=.85$), decision episodes in the later periods are significantly longer than the earlier ones ($F=11.73$, $df=2$; $p<.001$)

A contrast between the two projects is also evident when we compare participation in decision episodes. Table 3 shows that in Fire, the average number of individuals participating in a decision episode shrinks from 3.4 to 3.2 over time. It is apparent that this drop in participation occurs in core members (administrators) rather than in users.

In MSN, on the other hand, average participation steadily increases over time. In this project, the growth in participation is attributable to developers and users, while participation by administrators remains relatively constant.

| | Table3. Average Participants per Episode | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Beginning** | | | | **Middle** | | | | **End** | | | |
| **#** | **Admin** | **Dev** | **User** | **Total** | **Admin** | **Dev** | **User** | **Total** | **Admin** | **Dev** | **User** | **Total** |
| **Fire** | 1.3 | 0.7 | 1.35 | 3.35 | 0.6 | 1.1 | 1.55 | 3.25 | 0.6 | 1.2 | 1.4 | 3.2 |
| **aMSN** | 2.35 | 0.05 | 0.2 | 2.6 | 1.35 | 1.4 | 0.35 | 3.1 | 1.95 | 3.15 | 1.25 | 6.35 |

One-way ANOVA shows that the involvement of participants is relatively level overall in Fire over time (Number of total participants: $F=.05$, $df=2$; $p=.95$)   although participation by Administrators shrinks and participation by developers rises (Number of Administrator: $F=13.49$, $df=2$; $p<.001$, Number of developer: $F=.2.04$, $df=2$; $p=.14$; Number of user: $F=.22$, $df=2$, $p=.80$.)

However, in aMSN, the total number of participants in decision-making episodes increased dramatically ($F=14.536$, $df=2$; $p<.001$), mainly due to increasing involvement of developers and users (Developer: $F=29.21$, $df=2$; $p<.001$; user: $F=6.62$, $df=2$; $p<.01$). And we also see significant reduction in Administrators' involvement ($F=8.48$, $df=2$; $p<.01$).

These comparisons suggests that the two projects' decision processes (a) had significant differences, and (b) evolved differently over the sampling intervals. Over time, Fire's decision episodes gradually began to take longer, contain fewer messages, and include fewer participants, especially fewer Administrators.  aMSN on the other hand made decisions relatively faster, even when the participation of developers and users was growing. These patterns indicate increasing interest, energy, and inclusiveness in the decision processes at MSN, while Fire's patterns indicate a leveling off of interest and energy, especially among Administrators.

### *Who Triggers Decision Episodes and Announces Decisions?*

In order to understand the role played by core and peripheral members of the projects in creating decision opportunities for the group, we asked who sent the message that triggered  decision episodes, and who sent the message(s) that announced decisions   Table 4 shows that all three groups played a role in triggering decision opportunities, but that there were differences between projects, and from time period to time period.

| | Table 4.  Who Triggers Decision Episodes | | | |
|---|---|---|---|---|
| | | **Admin** | **Developer** | **User** |
| **aMSN** | **Beginning** | 12 | 0 | 8 |
| | **Middle** | 5 | 10 | 5 |
| | **End** | 8 | 9 | 3 |
| **Fire** | **Beginning** | 8 | 2 | 10 |
| | **Middle** | 0 | 3 | 17 |
| | **End** | 0 | 6 | 14 |

There are significant relationships between sampling period and the person triggering episode in both cases (aMSN:$X^2$=14.91, *df*=4, *p*<.01; Fire: $X^2$=20.17, *df*=4, *p*<.01)  The opportunities for decision-making in aMSN were originally initiated by Administrators and Users. In later periods, Administrators and users decreased their activities and more developers became involved in this process, suggesting the growth of a an energized core of developers.

In Fire, Administrators originally triggered opportunities, but dropped out later. Users continued to play a very important role in identifying problems and creating opportunities for decision-making, while Developers took up some of the slack for Administrators.

Table 5 shows that users rarely announce decisions in either project.  There are significant relationships between sampling period and the person announcing decision in both cases (aMSN:$X^2$=19.06, *df*=4, *p*<.01; Fire: $X^2$=21.04, *df*=4, *p*<.01), suggesting that change was taking place in both, primarily due to increasing activity by developers.  In aMSN, administrators remained active, while in Fire they did not.

| | | Admin | Developer | User |
|---|---|---|---|---|
| **Table 5.  Who Announces Decisions** | | | | |
| | | **Admin** | **Developer** | **User** |
| **aMSN** | **Beginning** | 21 | 0 | 0 |
| | **Middle** | 28 | 6 | 0 |
| | **End** | 20 | 11 | 5 |
| **Fire** | **Beginning** | 30 | 4 | 0 |
| | **Middle** | 13 | 17 | 0 |
| | **End** | 9 | 13 | 1 |

### *Decision Process Complexity*

We are especially interested in those episodes that are not straightforward or linear in nature.  These "Multiple-complex" episodes appear to unravel in ways that are consistent with the garbage-can decisions described by Cohen *et al.* where some problems (triggers) remain unresolved, while a variety of new choice opportunities may arrive late in an episode, and attach themselves to existing decision that seem to be waiting for them to appear.

Figures 2 and 3 show the evolution of decision complexity in Fire and aMSN respectively.  A marginally significant relationship existed between sampling period and process type in both cases (Fire: $X^2$= 8.00, *df*=4, *p*=.09; aMSN: $X^2$= 8.68, *df*=4, *p*=.07). Fire became more single focused, while aMSN became more of a "garbage can," with the number of Multiple-Complex episodes increasing. Future analysis will look for causes of this increase.

**Figure 2. Decision Complexity in Fire**



**Figure 3.  Decision Complexity in aMSN**
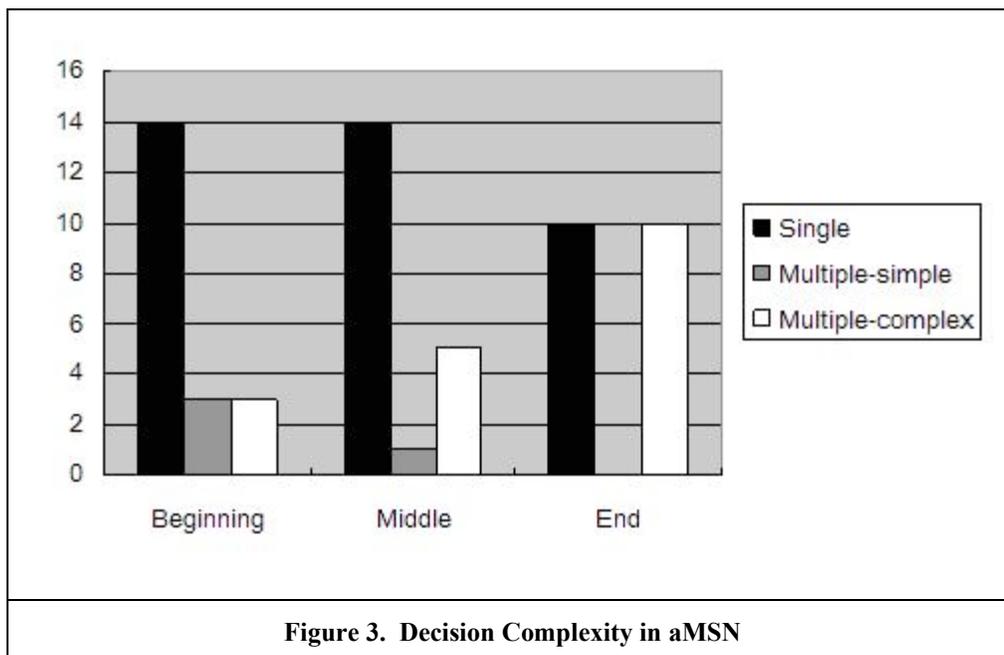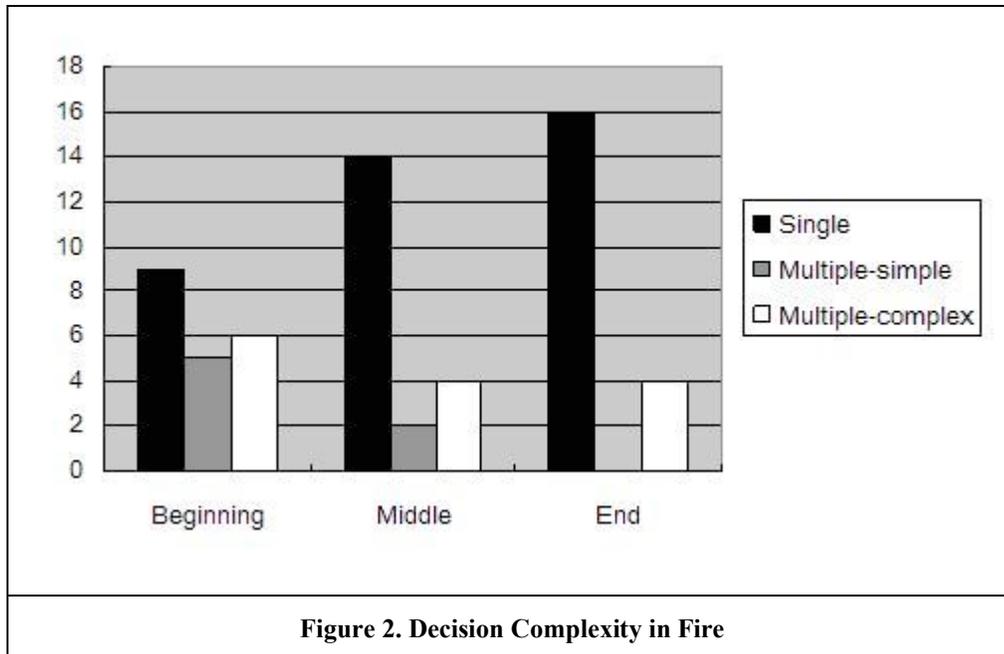
## Discussion and Conclusion

We have presented preliminary findings from a long term research project that seeks to identify work practices that make technology-supported self-organizing distributed teams effective in producing outputs, meeting member needs and continuing to function. This paper compared decision making practices from two open source project teams, and preliminary findings suggest that there may be important differences in their practices.  By sampling decision

episodes over time, we have been able to demonstrate that each project present an identifiable trajectory – one that connotes acceleration and energy (growth in number of participants, shorter decision time, more inclusive participation) and one that shows signs of deceleration and entropy (shrinking participation, disappearing leadership, longer decision cycles.)  We believe that these observations will prove to be related to a full multivariate measure of project success that includes not only downloads, but also development activity, spin-offs, and ability to attract and retain members.

But because this research is still in progress, it is premature to draw strong conclusions about the meaning of these observations.  We have much analysis still to do.  For example, we have yet to develop a validated typology of decision styles, and we have yet to fully relate the role of individuals to trigger types.  We have yet to apply the tools developed in this preliminary work to the other four projects we have selected.  By looking at the enterprise projects we will be able to observe the effects of different software context on work practices. But while there is still much to be done, we believe that the elementary variables we have identified to date will provide the foundation for a more in-depth qualitative analysis that will provide richer explanations of the patterns this quantitative content analysis reveals.

At the conference in December, we will have completed the analysis of all six projects, and will present a comprehensive analysis of the emergent decision making practices of technology-supported self-organizing distributed teams.

## References

Berdahl, J.L. "Gender and leadership in work groups: Six alternative models," *Leadership Quarterly* (7:1) 1996, pp 21–40.

Crowston, K., Howison, J., and Annabi, H. "Information systems success in Free and Open Source Software development: Theory and measures," *Software Process—Improvement and Practice*) In press.

DeSanctis, G., and Gallupe, B. "A foundation for the study of group decision support systems,"  (33:5) 1987, pp 589–609.

Drucker, P. *Managing the Non-Profit Organization: Principles and Practices* Harper Business, New York, 1990.

Duarte, D.L., and Snyder, N.T. *Mastering virtual teams*, (2nd ed.) Jossey-Bass, San Francisco, 2001.

Eden, C., and Ackermann, F. "Group decision and negotiation in strategy making," *Group Decision and Negotiation* (10:2) 2001, pp 119–140.

Erenkrantz, J.R. "Release Management Within Open Source Projects," Proceedings of the ICSE 3rd Workshop on Open Source, 2003.

Fitzgerald, B. "The transformation of Open Source Software," *MIS Quarterly* (30:4) 2006.

Fjermestad, J., and Hiltz, S.R. "An assessment of group support systems experiment research: Methodology and results," *Journal of Management Information Systems* (15:3) 1998/1999, pp 7–149.

German, D.M. "The GNOME project: A case study of open source, global software development," *Software Process: Improvement and Practice* (8:4) 2003, pp 201–215.

Grant, R.M. "Toward a knowledge-based theory of the firm," *Strategic Management Journal* (17:Winter) 1996, pp 109–122.

Hackman, J.R. "The design of work teams," in: *The Handbook of Organizational Behavior,* J.W. Lorsch (ed.), Prentice-Hall, Englewood Cliffs, NJ, 1987, pp. 315–342.

Halloran, T.J., and Scherlis, W.L. "High Quality and Open Source Software Practices," Meeting Challenges and Surviving Success: 2nd ICSE Workshop on Open Source Software Engineering, Orlando, FL, 2002.

House, R.J., and Aditya, R.N. "The social scientific study of leadership: Quo vadis? ," *Journal of Management* (23:3) 1997, pp 409–473.

Moon, J.Y., and Sproull, L. "Essence of distributed work: The case of Linux kernel," *First Monday* (5:11) 2000.

Pavitt, C. "Small Group Communication: A Theoretical Approach (3rd Ed)," 1998.

Pearce, C.L., and Conger, J.A. (eds.) *Shared Leadership. Reframing the Hows and Whys of Leadership.* Sage, Thousand Oaks, CA, 2003.

Raymond, E.S. "Homesteading the noosphere," *First Monday* (3:10) 1998a.

Raymond, E.S. "The cathedral and the bazaar," *First Monday* (3:3) 1998b.

Turoff, M., Hiltz, S.R., Bahgat, A.N.F., and Rana, A.R. "Distributed group support systems," *MIS Quarterly* (17:4) 1993, pp 399–417.

Watson-Manheim, M.B., Chudoba, K.M., and Crowston, K. "Discontinuities and continuities: A new way to understand virtual work," *Information, Technology and People* (15:3) 2002, pp 191–209.

Wayner, P. *Free For All* HarperCollins, New York, 2000.