MIS Quarterly

RESEARCH ARTICLE

MUTUAL UNDERSTANDING IN INFORMATION SYSTEMS DEVELOPMENT: CHANGES WITHIN AND ACROSS PROJECTS

Tracy A. Jenkin and Yolande E. Chan
Smith School of Business, Queen's University,
Kingston, ON CANADA K7L 3N6 {tracy.jenkin@queensu.ca} {ychan@queensu.ca}

Rajiv Sabherwal
Sam M. Walton College of Business, University of Arkansas,
Fayetteville, AR 72701 U.S.A. {rsabherwal@walton.uark.edu}

Appendix A

Proposed Sensegiving and Sensemaking Potential of Project Planning and Control Mechanisms

Mechanisms		Sensegiving (Actor: Sensegiver)	Sensemaking (Actor: Sensemaker)
Planning Mechanisms	Comprehensive	**High** In comprehensive planning processes, there tends to be a clear project vision developed, with the high-level details known up front: what, how, and why. The sensegiver provides these details to the sensemakers, typically as directives. Thus, the potential to give sense with this mechanism is high.	**Low** The sensemakers are given the vision and project details by the sensegiver(s). With these details in hand there is little uncertainty, so the sensemakers have little additional analysis and interpretation to do to gain an understanding. With the interpretation given, the potential for sensemaking is low. That is, not much sensemaking needs to be done.
	Emergent	**Moderate** Although few details (i.e., what, how, and why) are known up front and uncertainty is high, the planning process is iterative and techniques are very interactive between project stakeholders. Artifacts such as prototypes and storyboards are created and used to ultimately give sense. Uncertainty is reduced by such artifacts. Thus, the potential to give sense with this mechanism is moderate.	**High** There are few details known up front, with a high level of uncertainty. There is a high degree of interaction between sensegivers and sensemakers to analyze, discuss, and anticipate the project details. This is often a very iterative process. Therefore, the sensemakers must actively engage in dialogue and with artifacts to make sense and flesh out the details. Thus, the potential for sensemaking is high.

Mechanisms		Sensegiving (Actor: Sensegiver)	Sensemaking (Actor: Sensemaker)
Control Mechanisms	Self-Control	**Low** With self-control, there is only one individual involved. Thus, no one else is trying to influence that individual's behaviors and interpretation. Therefore, the potential for sensegiving is low.	**High** With self-control, there is only one individual involved. That individual is responsible for analyzing and interpreting the pertinent project details. Since no sense is being given, the individual has to flesh out all details and resolve any uncertainty. Thus, the potential for sensemaking is high.
	Clan Control	**High** The clan strongly influences the behaviors of its members through interactive socialization mechanisms. Thus, each member of the clan has the opportunity to act as the sensegiver of the clan's norms and goals. What (goal), how (behavioral norms), and why are often known and provided by the sensegiver(s). Sensegiving is also seen when clans reward appropriate behavior and sanction inappropriate behavior. Thus, the potential for sensegiving is high.	**High** The taken-for-granted nature of clan norms and goals results in little sensemaking. However, when clan norms and goals conflict with the project's norms and goals, sensemaking is required to resolve the resulting conflict. Clans have the ability to interact frequently, through multiple channels, so that this conflict can be resolved. Thus, the potential for sensemaking is high.
	Outcome Control	**Moderate** **Specifying outcomes**: Many details are specified by the sensegiver, typically the what (e.g., requirement) and why associated with the outcomes. The how, or process, may be left up to the sensemaker. There is some uncertainty and details to be determined by the sensemaker. Specification is typically accomplished using impersonal formats (i.e., artifacts). **Measuring outcomes**: The results being measured provide some details regarding the degree to which the performance metrics have been achieved. These are often high level and may require some analysis and interpretation (in this case, the sensegiver is the controllee who generates the results to be measured by the controller, the sensemaker). Thus, the sensegiving potential is moderate.	**Moderate** **Specifying outcomes**: The what and the why are typically provided (e.g., requirements document); however, the how is often left up to the discretion of the sensemaker. Thus, there is some uncertainty to resolve in terms of the process. The specification is typically provided as an artifact (e.g., document). **Measuring outcomes**: The sensemaker of the results (i.e., the controller) completes some analysis and reflection to determine the degree to which the results are meeting performance expectations. These results are high level and may require some analysis on the part of the controller (sensemaker). Thus, the potential for sensemaking is moderate.
	Behavior Control	**Moderate** **Specifying behaviors**: With behavior controls, sensemakers are told how to do their work (develop software), and often why it is important to do it that way. What to develop is not specified. It is up to the sensemaker to determine this. Specification of behaviors tends to be via impersonal formats (i.e., artifacts such as standards and methodologies). **Measuring behaviors**: Observation, status reports, and gate reviews provide some information indicating whether the process is being followed and if not, why. The controllee is the sensegiver of this high-level information. Thus, the sensegiving potential is moderate.	**Moderate** **Specifying behaviors**: Although "how" to do the work and often "why" it is important to do it that way are given to the sensemaker, (s)he is not told specifically what to develop. The sensemaker is left to determine this aspect. Also, adaptation of the specified behaviors is often acceptable and requires interpretation on the part of the sensemakers to determine. **Measuring behaviors**: The sensemaker of the given reports, verbal status, or observations has to assess the extent to which the behavior is as specified. The sensemaker is the controller. Not all details are provided in reports, nor can all behaviors be observed. Therefore, there is some level of uncertainty requiring analysis and interpretation by the sensemaker. Thus, the potential for sensemaking is moderate.

Appendix B

Linkage between ISD Project Planning and Control Mechanisms and Sensemaking and Sensegiving

Mechanisms		Sensegiving	Sensemaking
Planning	Comprehensive	"Information *analysts then construct* alternative structures for the overall MIS architecture subject to the MIS objectives, strategies, and constraints enumerated as the MIS strategy set. The general *alternatives are then presented to management*" (Bowman et al. 1983, p. 17).	"When the new members officially joined the team, they were given copies of deliverables from the planning phase. In *a 'knowledge dump' meeting*, the planning phase *strategists walked the newcomers through the planning phase deliverables* presentation" (Levina 2005, p. 121).
	Emergent	"Structured problem solving and experimentation through the use of cross-functional teams tend to produce *high levels of understanding regarding organizational processes ... a noted drawback of the philosophy is its tendency to promote strategic drift as many actors continually bounce back and forth between competing strategic perspectives*" (Segars and Grover 1999, p. 222).	"As the third week of the planning phase approached—a time when, according to the Eserve service delivery model, consultants should be up to speed on their client's business—Eserve *conducted a brainstorming workshop with the intention of generating ideas* as to what kind of functionality the website should support" (Levina 2005, p. 119).
Control	Self-Control	"Self-control therefore requires that the controller grant *autonomy to the controllee without imposing any other forms of control*" (Tiwana and Keil 2009, p. 21).	"He *monitors how well he is progressing* over time; and *intrinsically rewards himself* for completing the job" (Kirsch 1996, p. 3).
	Clan Control	"Team-building sessions and meetings *served to incent stakeholders to embrace common values*. The System Development Manager explained various approaches for *generating this level of commitment and shared goals*" (Kirsch 2004, p. 383).	"During this phase, control can be characterized as *'collective sensemaking'* in which informal mechanisms of control are used jointly by IS and business stakeholders to *clarify ambiguous project goals, reach consensus on a common business process, and negotiate a set of global system requirements*" (Kirsch 2004, p. 388-389).
	Outcome Control	"In outcome control, the controller focuses on the outputs (both final and interim) of the project *without regard to the process by which these outputs are achieved*" (Choudhury and Sabherwal 2003, p. 293).	"that define outcome metrics for software in great detail (high standardization of performance criteria). However, the contracts allow project teams to have *discretion in adopting the standards* to suit the context of specific projects" (Nidumolu and Subramani 2003, p. 168).
	Behavior Control	"A detailed systems development methodology may be viewed as a mechanism of behavior control if it articulates *the precise steps to follow to successfully develop a system*" (Kirsch 1997, p. 217).	"While younger consultants … tended to interpret the methodology literally, and follow its procedures 'to the letter,' … experienced employees seemed less constrained, *relying on their initiative to direct production work, and using the methodology primarily as a coordination device to manage impressions* through appropriate behavior packaging. Thus, while the methodology is prescriptive in documentation, in practice, its tenets were often modified, overridden, or ignored" (Orlikowski 1991, p. 16).

Legend: Low Moderate High

Appendix C

Research Method Summary: Building Theory from Cases (Eisenhardt 1989)

	Recommendation	This Study
Planning the Study	**Getting started**: Define the research question and identify *a priori* research constructs.	Used a combination of top-down inductive and deductive theorizing to start the study as recommended by Shepherd and Sutcliffe (2011), followed by bottom-up inductive theorizing described below in this appendix.
	Selecting cases: Identify the population of interest and use theoretical sampling.	Defined population as projects focused on the development and enhancement of systems at global IT firms with headquarters in North America. Sample included projects completed over a 10-year time span (2000 to 2010) at two organizations (Alpha and Beta). Examined 13 projects (7 at Alpha, 6 at Beta).
Collecting Data	**Crafting instruments and protocols**: Use multiple data collection methods, combine qualitative and quantitative data, and involve multiple investigators.	Developed and refined interview guide before the interviews. Most data were qualitative, although some quantitative data were also captured (e.g., team members per project, years of experience for each team member). The authors jointly developed the interview guide. One author conducted the interviews and immersed herself in the case details and project context, enabling the other authors to bring a "very different and possibly more objective eye to the evidence" (Eisenhardt 1989, p. 538).
	Entering the field: Overlap data collection and analysis, and use flexible methods to collect data.	Data collected in 2004, 2005, and 2010, through 24 interviews with 21 informants at the two organizations. Data collection and analysis overlapped over this time period. For instance, the authors jointly considered the interview findings (in the form of narrative summaries) to plan subsequent interviews and data collection. At least one respondent at each organization was interviewed in multiple time periods. Interviews were recorded, producing 214 pages of transcripts.
Developing Propositions	**Analyzing data**: Analyze within-case data and search for cross-case patterns.	Qualitatively coded quotes from transcripts based on the constructs in the initial model (Figure 1) and the project stage. Used iterative process with multiple cycles with all authors involved. Next, used coded transcripts to rate each project's stages in terms of the focal constructs in the initial model as high, moderate, or low. Also identified planning and control mechanisms that were used in each project stage to code and calculate sensegiving (sensemaking) potential. In the final phase, identified sensegiving-sensemaking episodes, coding when it occurred, the stakeholder groups who gave and made sense, and directionality. Result was panel data for 13 projects at three points in time. Created multiple data displays for within and cross-case analysis. Regressions using the quantitative data supplemented the qualitative results.
	Shaping hypotheses: Iteratively compare theory and data so that emergent theory fits the data.	Employed a longitudinal embedded mixed-methods design approach (Creswell and Clark 2007). Used regressions to develop an initial understanding of the potential relationships. Used the extensive qualitative data to develop more nuanced insights into various relationships. Triangulation through the use of quantitative data (regressions), qualitative data (the coded text from the raw transcripts), and overall perceptions (narrative summary, data displays in Figures 3, 4, 5, and 6).
	Enfolding literature: Compare theory and hypotheses with the literature.	Compared the patterns and relationships uncovered through the qualitative analyses with the initial model, the prior literature, and the quantitative analyses. Developed an emergent model and propositions based on these new relationships and patterns.
	Reaching closure: Achieve saturation (i.e., minimal improvement).	Reached closure in data analyses once the emergent model and propositions were consistent with the data (including the transcripts, key informants' remarks, and quantitative results), and the incremental improvement from continuing the analysis or examining other projects seemed minimal.

Appendix D

Supplementary Project Details[1]

Proj.	Background	Size[2]	Novelty	Key Events[3]	Key Outcomes
A1	New feature requested by customers, referred to as a "customer escalation."	3 to 5	• New technology for Alpha • New product feature	• Quality assurance (QA) finding issues during testing was the initial stimulus for the team thinking there were issues. • Customer reports of significant issues with the software confirmed that there were major issues with the new feature and technology. • Post-implementation discovery that other users of this IT had similar issues.	• Late delivery with a number of defects, resulting in customer complaints. Followed up with patches to provide short-term fixes.
A2	Architecture project to introduce new IT and structure to an old legacy product. This would enable future user interface enhancements. Proposed by team lead.	6 to 8	• Existing product functionality implemented on new technology platform	• Prototyping to assess technical feasibility and whether to proceed with the project. • Decision to proceed based on successful prototype.	• Successfully implemented architecture changes and fixed a number of long-standing defects in the product.
A3	Initiated to resolve the issues associated with the new feature implemented on A1.	4 to 7	• Technology introduced on A1 • Product feature introduced on A1	• Decision to consult and work with another office (in Alpha) and reuse one of their code components. • Issue with Windows disrupted development for four weeks. • Cut scope back due to time constraints.	• Successful delivery that resolved problems in A1 and met customer needs.
A4	Existing functionality from a 15-year-old product was being incrementally ported to a web-based platform in multiple projects, including A4. Identified as important by sales and marketing.	6 to 9	• Existing product functionality implemented on new web-based platform • New methodology introduced	• Usability testing of prototype at a user conference found usability issues before development began. • Decision to split off functionality to create project A5.	• Successful delivery of product to customers, with minimal defects. Held up as the ideal way to run a project, but delivered later than target date due to problems with A5.
A5	Originally part of the requirements of A4, but spun off as a separate project given the scope of the required functionality. A4 was dependent on the functionality being implemented in A5.	6 to 9	• New product features for existing platform and new web-based platform • New methodology recently introduced, but not applied here	• Decision to split off functionality to create project A5. • Realization that scope was larger than initially estimated. • Tester posed questions seeking clarifications regarding user experience document, which initiated sensegiving-sensemaking episodes to further define requirements. • Decision to split release of the product to provide select customers with an early version of the product (with defects).	• Delivered required product functionality late and with defects. Split release allowed some customers to receive functionality earlier than others, based on urgency of needs. • Failed to implement methodology and lessons from A4. Significant problems with the process led to lessons learned at the end of the project.

[1]For timeline information, see Table 2.

[2]Size is in terms of the number of individuals working on the project, which changed over time. In addition to the IT project team members, it includes the business sponsor, the business project manager, any business-side team members, and the vendor project manager as applicable.

[3]These are the significant events in the project that impacted the project or subsequent projects.

Proj.	Background	Size	Novelty	Key Events	Key Outcomes
A6	Involved implementing several "specific" features to Alpha's flagship product. The goal was for customers to switch to it from legacy product. Considered a point release. Not all features were relevant to all customers.	14 to 16	• Minor feature updates to existing product and platform	• Change in product manager midway through the project. • Decision to drop large, more complex feature to meet deadline. • Decision to release patches to fix issues post-release after customers called to complain.	• Product was implemented on time, but with some quality issues. Specifically, customers were not entirely satisfied with the product because some older problems still existed. • Patches were released to fix these issues.
A7	A major overhaul of Alpha's flagship product (same as A6), including new features and improvements to existing features. Also added integration with Alpha's suite of products. Considered a full release.	14 to 16	• New features and feature updates to existing product and platform • New methodology introduced	• Organization-wide introduction of new methodology, creating turmoil on project. • Identified and agreed upon adjustments to new methodology. • Decision to drop large feature (same feature dropped in A6).	• Dropped feature and turmoil at project start hurt perceptions of success. • Otherwise, product was successfully launched. • Process improvements and adaptation of the new methodology.
B1	Implemented a new product and new technology platform and business processes to support it. Considered strategic, with product launch seen as key to the company's survival.	20 to 30	• New technology platform, consisting of multiple new and existing systems • New business processes	• Period of high intensity of change requests, resulting in significant turmoil on the project. • Decision to implement the project according to scheduled release date.	• Delivered project on time but with numerous defects. Turmoil and defects negatively impacted perceptions of success. • Did achieve the objective of launching new product.
B2	Originally part of project B1, involved porting legacy application to the new technology platform implemented in B1.	20 to 30	• Existing legacy system features implemented on IT platform introduced in B1	• Late change request from critical business partner. • Decision to postpone project by one month to accommodate request.	• Delivered project one month later than original plan, due to postponement.
B3	Considered a strategic project that involved insourcing an existing customer product that was previously outsourced.	15 to 22	• New features on existing IT platform • New business processes linked to the new product	• Programmers hid technical issues. • Decision to reduce scope late in the project due to technical issues, which had not been disclosed earlier.	• Significantly over budget, late and defect-ridden. Technical issues resulted in reduced scope and manual workarounds.
B4	Added functionality to enable business units to modify own (customer facing) applications and accelerate implementation of business initiatives.	20 to 25	• New features and processes on existing technology platform	• Programmers hid technical issues, just like in B3. • Decision to drop a large portion of project's scope at the last minute due to technical deficiencies, which had not been disclosed earlier. Just like in B4.	• Due to scope reduction, project objectives not met. Business stakeholders were not satisfied. • A new project was initiated.
B5	Focused on improving global procurement and order management systems, information, and processes.	15 to 20	• New third-party technology platform • New business processes • Changes to methodology	• Implemented request by programmers to change process for managing issues. • Process changed back when it did not have the intended effect. • Change in methodology declared by senior management.	• Project successfully delivered the desired functionality for the business. Business and IT stakeholders perceived project as successful.

Proj.	Background	Size	Novelty	Key Events	Key Outcomes
B6	Involved implementing functionality and processes that would allow a business unit to expand its marketing capabilities. Similar capabilities, systems, and processes were implemented in another unit.	17 to 27	• New features in an existing IT platform • New business processes • Similar to processes and features in a related system	• As requested by the project manager, business management agreed to turn off old system so that the new features did not have to be implemented on it. • Reversal of this decision at project end when senior sales reps became aware of the impending system retirement. • Post-implementation workarounds.	• The issue with the senior sales reps resulted in significant workarounds after go-live. All other aspects of the project were considered successful. Not all stakeholders perceived the project as a success.

Appendix E

Interview Guide

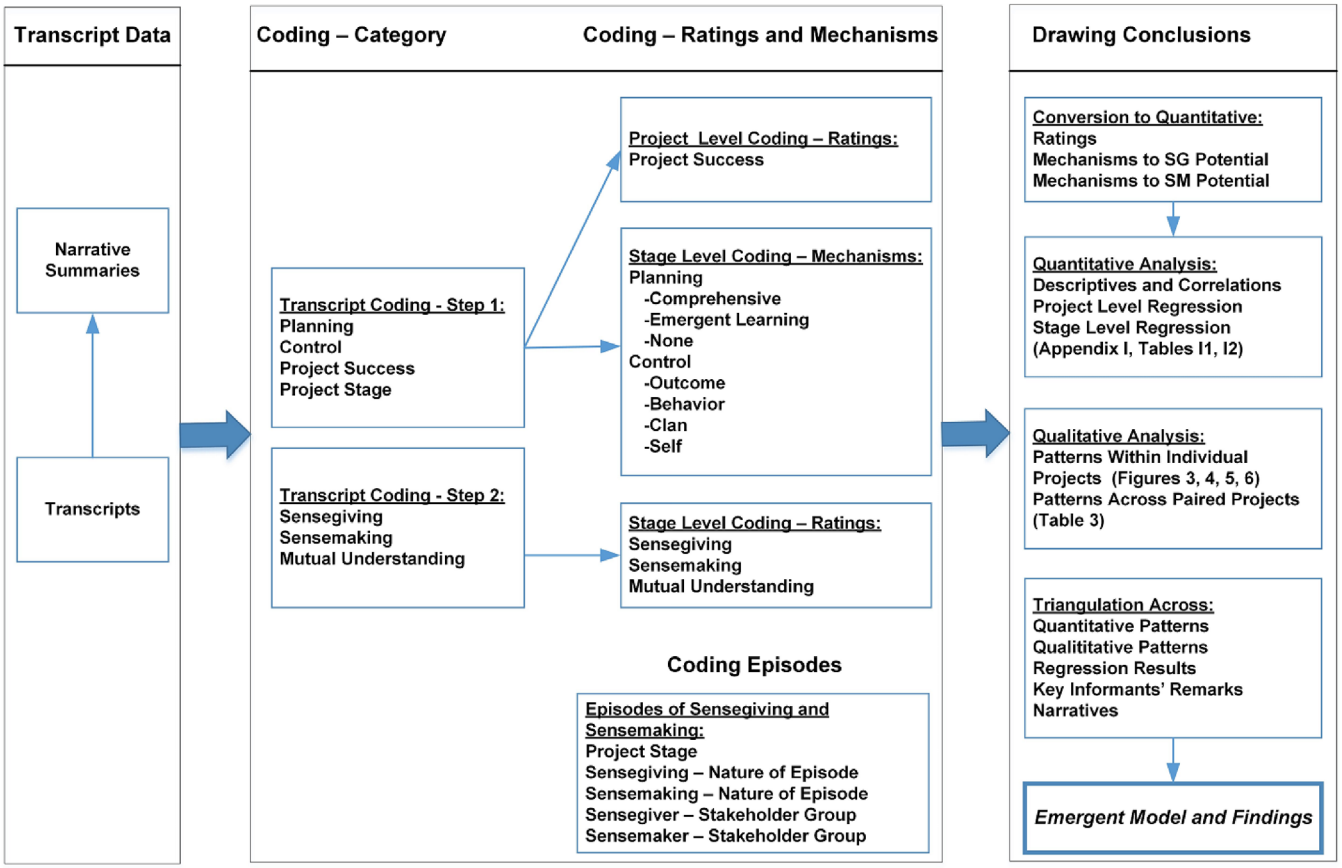
The following questions were used as a guide during the interview. Each interview was emergent: the interviewer adapted questions and asked additional probing questions as needed.

Please recall a recently completed project that was considered strategic (important to the business). We will discuss this project throughout the interview.

1. What is the name of the project?
2. What was your role on the project?
3. How long have you been with the company? In this role?
4. When did the project start and end?
5. Was there a business sponsor? (Probe: Ask about steering committee and other senior management support mechanisms.)
6. Can you describe the project organization chart? Is this typical for the organization?
7. Why was this project considered strategic (important to the business)?
8. Do you know what priority level was assigned to this project?
9. Did this project need to be justified formally before it was approved? If so, what justifications were provided?
10. What were the purpose, goals, and objectives of the project? (Probe: Discuss linkages or dependencies on other systems or projects.)
11. What changes occurred during the project?
12. At any time, was this project's priority changed?
13. At any time were there resource changes on the project? If so, why were those changes made?
14. Were there any changes to requirements during the project? How were changes in requirements handled?
15. Were you given adequate ownership and decision-making ability in running this project? (for Project Manager)
16. What do you think makes a great project manager?
17. How did you handle roadblocks (problems, unexpected issues, surprises)? Did you have predetermined escalation paths? (Probe: Discuss risks and risk management approach; also discuss any major issues and why they arose.)
18. What, if any, methodology was followed? Describe. How closely did you follow it? (Probe: Describe the phases and what was done in those phases. Specifically, initiation, planning, design, development, testing and implementation phases.)
19. At a high-level, explain your testing methodology. (Probe: Describe any problems detected during testing, such as defects.)
20. Were periodic reviews of the project conducted? (Including testing, gate reviews, deliverable reviews, status reviews, etc.)
21. Was a final evaluation of the project conducted? What were the criteria used for the evaluation? What was done with the results? (For example, formal post-mortem review with the business? Surveys?)
22. Was a formal process followed for the implementation (deployment to "production" or "generally available") of this project? If yes, what was it? If no, what was involved in coordinating the implementation of this system?
23. Most important best practice? Biggest mistake?
24. How did the business (for example, product managers, marketing, sales) communicate with the project team? How did the project team communicate with the business? (Probe: Were there any problems with communication and coordination? How were users involved?)
25. How would you describe the relationship you had with the business? (for development team members) How would you describe the relationship you had with the development project team members? (for business team members) (Probe to see if there were there any problems with the relationship.)
26. In your opinion, how well do the development project team members understand the business? Conversely, how well do the business team members understand development?
27. Was the project implemented on time? Did this project have an accelerated timeline to meet market-timing constraints? Or was timing based on estimated effort?
28. Was the project implemented on budget?
29. Was this project considered successful? Why or why not?
30. How does your company measure the success/performance of a project?
31. Did the initial objectives and goals for the project change during the project? Were all of the objectives and goals of the project met?
32. Were some objectives for this project not met? Was this a conscious decision (for example, ran out of time or money)? If some objectives were not met, what was done as a result?
33. Was the business (and users) satisfied with the results of the project?
34. Was the development team satisfied with the results of the project?

Appendix F

Coding and Analysis Approach



Appendix G

Quotes Illustrating the Coding of Constructs

Construct	Type	Illustrative Quote	Project (Stage)
Planning	Comprehensive	"Because of the high risk of this project, I had put together an MS Project plan with the team lead. We mapped out the whole process and defined the details" (Project Manager).	A2 (E)
	Emergent	"We cut back a little on what we were going to deliver." When asked if this was done because of timing, he responded, "Yes. Because of scope. So we decided to do it in a phased delivery of functionality" (Project Manager).	A3 (L)
Control	Self-control	"[The initial document] was a high-level vision. This is how we see it happening. The developer thought that it was exactly what should be happening and then started developing and there wasn't time to go to a more detailed design of here is how the screens will actually look" (Product Designer).	A5 (E)
	Clan control	"No training, no rollout, no explanation. Half of the documents are gone, or the documents we had been using have no meat in them anymore. I went out to look at the test spec and it was 'huh?' And people came to me asking me if I had an old copy of the template. And I say 'yes we have it on our Sharepoint site'" (QA Manager).	B5 (M)
	Outcome control	"Reviews were being done, but primarily between the development manager and the product manager in terms of 'have you delivered this yet.' Very centralized" (Product Designer).	A6 (M)
	Behavior control	"We had weekly reviews of the actual specifics of the project in the Exec Sponsor's project review meeting. So any large efforts that are being worked in his group are talked about in this forum—a weekly Friday meeting with all of his direct reports and stakeholders throughout the business that own pieces of his business (e.g., risk manager). That was his forum for Q and A, if he had questions about what we were doing or we needed answers for something, we would bring those up in that meeting" (Business Project Manager).	B6 (M)
Sensemaking	Low	"I did find there was a lot of coming and saying 'this is what I am getting' and then me saying 'well that is what you are supposed to get. It is by design. If you had read the document you would know that'" (Programmer 2).	A5 (M)
	High	"Having a dedicated product designer, a usability person/product designer, to help define the stories and then having review sessions with the team where people can ask questions and change the stories if issues are brought up … took a lot of time, but was a really helpful process. And great for us (documentation). Also good for the product" (Technical Writer).	A7 (M)
Sensegiving	High	"The business area originally felt they could wait until after implementation to address their needs … resulted in a last-minute escalation to the project steering committee and the CEO .... Very political move" (IT Project Manager).	B2 (L)
	Low	"We were compliant … in form, but not in function. There was a lot of spin going on. IT couldn't speak to the business truthfully" (QA Manager).	B3 (M)

Construct	Type	Illustrative Quote	Project (Stage)
Mutual Understanding	High	"Everybody knew what everybody else was going to do and how it is going to work in the system" (Team Lead).	A4 (E)
	Low	"At the beginning, there was an organization-wide decision … that generated a lot of confusion and lack of clarity within the team.… this team, as well as other teams within the organization, were led to question some fundamental things about the way they were working" (Product Designer).	A7 (E)
Project Success	Low	"In the end, it didn't really accomplish what it was intended to do. The users weren't that satisfied" (Project Manager).	A1 (L)
	High	"That one went quite nicely. It was one of those classic 'yeah that's the way you do it'" (Product Designer).	A4 (L)

Appendix H

Quotes Illustrating the Coding of a Project: Project A7 [redacted]

Stage	Early	Middle	Late
Planning (overview)	The requirements for the project were determined by the product manager at the start. Planning was not as inclusive as it could be given that customers were not consulted. However, planning was disrupted in the early stages of the project because of the methodology change. Once this change was resolved and planning resumed, planning was conducted collaboratively and emerged incrementally during each iteration.		
Coding	**Comprehensive**	**Emergent**	**Emergent**
Illustrative Quotes	"They had a very ambitious set of objectives initially…. But very quickly that was taken off the list of goals because it was just too ambitious. I'm not sure how serious that goal ever was, but it was there. That was unfortunate but then they had a more realistic set of goals to replace that and I think it's been really successful, the process" (Technical Writer 1).	"A lot of collaboration between the new product manager, the dev manager and myself to set a common direction for the team, to really agree on the priorities for the release and to react very flexibly to changes in priority. So that has been very effective" (Product Designer).	"But lately it is more visible because every two weeks we get together to discuss what's new" (Programmer).
Explanation	Product manager set the list of ambitious objectives upfront on the project. These objectives were updated to become more realistic. Demonstrates comprehensive planning upfront. Other comments also show that all the requirements were determined upfront, but things were added and taken away during the agile iterations.	Collaboration between the product designer, development manager, and product manager show that these three stakeholders helped to set priorities and update those priorities as needed when changes arose. This began after things settled down following the methodology change. Other comments demonstrate that this collaboration enabled the team to adapt to changes flexibly across iterations, reflecting an emergent learning approach to planning.	Applies to both middle and late stages of the project. At the beginning of each iteration, the new things to be implemented in that iteration were discussed and planned in more detail. This demonstrates emergent planning in both the middle and late stages of the project.
Control (overview)	At the start of the project, there was little control. Clan control helped team members focus on overcoming the challenges associated with the methodology change. The new methodology was a form of behavior control, but it was not yet understood and was being compared to the old ways. Midway in the project, outcome controls were added: macro stories (requirements) served as outcome specifications and frequent testing and reviews served as outcome measurement. Behavior control was stronger now, as the methodology was more clearly understood. Daily reviews also helped measure behavioral compliance.		
Coding	**Behavior, Clan**	**Behavior, Outcome**	**Behavior, Outcome**
Illustrative Quotes	"Conflicts between product designer and developers early on over the new process because there was a new role for the designers and I think it was just a matter of sorting out what the roles would be. But there were some pretty significant conflicts" (Technical Writer 1).	"It helps things from getting too far off track. Because if someone has done something but maybe they haven't emailed you or let you know. But then they mention it and you know right away. Before when a developer changed something without telling	"Reviews were done at the end of each iteration. There would be a demo to interested parties as to what was achieved" (Programmer).

Stage	Early	Middle	Late
		you for whatever reason, it could be weeks or months before you found out about it. Whereas now, you could know the same day" (Technical Writer 1).	
Explanation	Conflicts were resolved using clan control, leveraging past history of norms and beliefs for the group, who had just completed project A6. Other quotes show that sanctions were imposed on members who did not accept new norms and roles.	Daily stand up meetings, which were conducted in both middle and late stages, served as an outcome control because they enabled outcome measurement. As noted above, this also enabled behavioral compliance to the adapted methodology. Other comments demonstrate the value of macro stories as outcome specifications.	Refers to the reviews and demos that occurred at the end of each iteration, in both the middle and later stages. These reviews served as a mechanism to measure outcomes. This demonstrates outcome control.
Cognitive Activities: Sensegiving (overview)	At the start of the project, there was moderate sensegiving from the espoused new methodology. This adapted methodology was a helpful sensegiving device throughout the rest of the project. Midway and later in the project, macro stories, interim builds (prototypes), daily stand-ups and frequent reviews gave the team sense, which in turn supported sensemaking.		
Coding	Moderate	High	High
Illustrative Quotes	"I think the introduction of the new process was a good stimulus for looking at some of the practices and improving the way the team does its work. And I think we have become more agile, not necessarily because of the introduction of the new process, but perhaps even in spite of it, by realizing that there were certain things that were important that we could be doing. It turned out to be a positive development definitely" (Product Designer).	"The other thing that was a best practice that emerged out of this release was early review of requirements and design with the dev team and with QA and documentation staff and even with support. So we have formalized a process, well not formalized, but we have a consistent process for doing these reviews and engaging the rest of the team in providing early and often feedback to priorities, well maybe not so much to priorities but definitely to design decisions. And that has been really key to effectively deliver the more complicated features we have in this release" (Product Designer).	Paraphrased: The amount of information and detail made writing up the documentation in parallel to development a lot easier. It allowed you to know what the functionality was going to look and act like when it was done. Things were completely itemized in detail and planned out (Technical Writer 2 Paraphrase: participant asked not to be directly quoted).
Explanation	Moderate sensegiving early in the project via the new methodology announcement. Other comments suggest that training and the project objectives provided sensegiving. However, sensegiving was only moderate, not high, because the methodology and training were too high level and not directly applicable to the Alpha and project A7 context. See sensemaking comments for how the team made sense of this.	A high level of sensegiving in the middle and later stages as the result of the reviews between the product designer and the development team, QA and technical writers and updating these stakeholders on design decisions and the rationale behind those decisions. Other comments also suggest that the macro stories (user experience stories) developed by the product designer to describe and explain the design to this same set of stakeholders provided a high level of sensegiving.	Demonstrates the sensegiving provided through the macro stories provided in each iteration in both middle and late stages of the project. This brought further sensegiving to the reviews, keeping a high level of sensegiving in both the middle and later stages. Other comments suggest that the daily stand up meetings, interim builds and demos enabled a high level of sensegiving.

Stage	Early	Middle	Late
Cognitive Activities: Sensemaking (overview)	At the start of the project, the level of sensemaking was high due to the team interpreting and adapting the methodology to suit their needs and resolving the uncertainty that came from the change. Midway and later in the project, macro stories, interim builds (prototypes), daily stand-ups and frequent reviews gave the team sense, which in turn supported sensemaking.		
Coding	**High**	**High**	**High**
Illustrative Quotes	"The way the process was rolled out was not done very well. As a result, this team, as well as other teams within the organization, were led to question some fundamental things about the way they were working. And there was quite a bit of churn and discussion that had to result before the team could settle back down and be effective in implementing the things they needed to implement" (Product Designer).	"I wouldn't say that was so much a feature of the formal process, but more of a feature of constant collaboration between what I like to call the customer team—so the product manager, the dev manager, and myself. As soon as time-sensitive things came up we immediately discussed them, we talked to the development team to understand the repercussions, got rough effort estimates and based on that quickly responded whether or not it was something that we could incorporate into the release or push off to a future release" (Product Designer).	"For A7, we have had the regular demos of the completed functionality for the whole team. And especially this was helpful in involving QA and documentation and support in understanding what had been delivered in each of the iterations preceding the demo" (Product Designer).
Explanation	A high level of sensemaking occurred early in the project as all team members tried to understand and work out how the new methodology could be applied to the project context of A7 and Alpha more broadly. Part of this sensemaking included reflecting on problems with the current process and how to adapt the new methodology to improve this process. Other comments demonstrate that the end result was an improved process.	Describes the sensemaking process engaged in by the "customer team" (product designer, development manager, and product manager) to understand impacts, develop estimates and make decisions. Along with the other comments that demonstrate the sensemaking that occurred via regular reviews, demos, and macro stories, this reflects a high level of sensemaking on a daily and weekly basis during the iterations.	A high level of sensemaking continued in the middle and later stages of the project. This comment demonstrates the sensemaking that the demos enabled for QA, technical writers, and the support team. Other comments demonstrate the benefit of regular reviews and macro stories in helping programmers, QA and technical writers make sense of the requirements as well as the functionality that had been implemented by programmers.
Cognitive Outcome: Mutual Understanding (overview)	At the beginning of the project, there was a low mutual understanding of the process and roles due to the change in methodology. Midway through the project, there was some uncertainty regarding why the large feature was dropped. However, the mutual understanding of the project process and the requirements progressively developed over the course of the project due to the high levels of sensegiving and sensemaking midway and later in the project.		
Coding	**Low**	**Moderate**	**High**
Illustrative Quotes	"At the beginning of [project A7], there was an organization-wide decision and a series of training sessions rolled out to support that decision to move to a more industry standard agile process. Not entirely industry standard but closer to those standards. That generated a lot of confusion and lack of clarity within the team" (Product Designer).	"And for IDP, you are supposed to have a post-mortem at the end of each iteration. So every three weeks or two weeks. I think those eventually died because no one said anything. Initially, it was good because I think when the process was still being nailed down it was helpful to be able to vent. But as people got accustomed to it, there was less need for that" (Technical Writer 1).	Paraphrased: in the new process, there are macro stories and micro stories provided and we have daily stand-up meetings. As a result, everyone on the team has a good sense of what is going on in terms of who is working on what, what is not working, and what is working well (Technical Writer 2 Paraphrase).

Stage	Early	Middle	Late
Explanation	The methodology change resulted in confusion regarding the impact of the change on roles and the process. Thus, there was low mutual understanding at the start of the project. Other comments further support the uncertainty and confusion that ensued in the early stage of the project while the team was working this out.	Post-mortems at the end of each two-week iteration were conducted initially when mutual understanding of the project and the product needed to be further developed. This mutual understanding grew to a moderate level during the middle of the project. Other comments note that a feature was dropped because of uncertainty regarding the required functionality. By dropping the feature, the team could focus on the better understood features.	Artifacts of the new process resulted in a high level of mutual understanding of the product being developed and the status of the project via the sensegiving and sensemaking enabled by these artifacts. In addition to the macro stories and daily stand up meetings, other comments demonstrate that regular demos and reviews with the entire team also helped in this regard.
Project Success	Moderate Success (success is evaluated at the end of each project).		
Coding	**Moderate**		
Illustrative Quotes	"I think A7 has done a much better job of addressing the strategic roadmap goals of the product. Again, one of the strategic roadmap goals was that one large feature that got dropped. At the same time, arguably, by not taking on that very complicated and risky feature, we have delivered a better and more stable product in a lot of other ways" (Product Designer).		
Explanation	Project stakeholders perceived the project to be moderately successful and not highly successful because of the important feature that was dropped from the project due to timing constraints. Other comments indicate that there was a gap created by the missing functionality, which impacted perceptions of success. Further, other comments demonstrate that the initial chaos caused by the methodology change had a negative impact on perceptions of project success.		

Appendix I

Quantitative Analyses and Results

We coded variables into low, medium, and high. Since we cannot assume equal gaps between low and medium and between medium and high, the data is ordinal in nature. Therefore, we used *Kendall's Tau* to measure correlation (Conover 1999) and conducted ordinal logistic regressions. Tables I1 and I2 present the descriptive statistics, correlations and the results for four regression models. All four regressions included two control variables: project duration (months) and number of primary informants for the project.

The first three regression models used stage-level data, with sensegiving, sensemaking, and MU as dependent variables. The planning and control mechanisms used in each stage were included at an aggregate level as sensegiving potential and sensemaking potential. Model 1 suggests that sensegiving during a stage depends on sensegiving potential and sensemaking during that stage. Similarly, Model 2 suggests that sensemaking in a stage depends on sensemaking potential and sensegiving in that stage. Thus, sensegiving and sensemaking affect each other, as expected. Moreover, sensegiving and sensemaking seem to depend on sensegiving potential and sensemaking potential, respectively, of the planning and control mechanisms used, as was also expected.

We expected both sensegiving and sensemaking to affect MU. The results for Model 3 differ somewhat from these expectations, suggesting that MU during a stage depends directly on sensemaking but *not* on sensegiving. Combined with the results for Model 2, this leads to the emergent proposition P1.

Model 4 used project-level data as project success, the dependent variable, was assessed at the end of the project. Although the sample size of 13 is a limitation and implied a low statistical power (0.36 with medium effect size), the effect of MU on project success was significant. Also, when sensemaking and sensegiving were added as independent variables, and a stepwise ordinal regression was conducted (due to the small sample size), only MU was found to have a significant direct effect on project success.[4]

Table I1. Descriptives and Correlations

					Correlations[1]				
	Mean	**S.D.**	**Min**	**Max**	**SG_P**	**SM_P**	**SG**	**SM**	**MU**
Sensegiving Potential (SG_P)	**0.46**	**0.25**	**0.00**	**0.88**					
Sensemaking Potential (SM_P)	0.37	0.25	0.13	1.00	0.18				
Sensegiving (SG)	2.18	0.72	1.00	3.00	0.28**	0.22*			
Sensemaking (SM)	2.00	0.79	1.00	3.00	0.09	0.28**	0.39***		
Mutual Understanding (MU)	1.97	0.78	1.00	3.00	-0.05	0.20*	0.26*	0.40***	
Project Success (SUCC)	2.15	0.80	1.00	3.00	0.27	0.31	0.49*	0.50*	0.53**

[1]Kendall's Tau statistics are reported for correlations. N = 13 for Project Success and correlations involving it. N = 39 for other constructs and correlations. *p < 0.05; **p < 0.01; ***p < 0.001. As expected (Conover 1999), the results are consistent in significance when using *Spearman's Rho*.

[4]The results for this additional regression are not reported in Table I2 due to space constraints. Moreover, sample size limitations precluded quantitative tests of the mediating effects.

Table I2. Regression Results[1]

	Model 1[2]	Model 2	Model 3	Model 4[3]
Dependent Variables ➔	**Sensegiving**	**Sensemaking**	**Mutual Understanding**	**Project Success**
Control Variables				
Project duration (months)	0.00 (0.82)	-0.12 (0.10)	-0.34 (0.22)	-0.01 (0.11)
Number of primary informants for the project	-0.47 (0.17)	0.40 (0.33)	0.61 (0.39)	-0.27 (0.37)
Independent Variables ⬇				
Sensegiving		2.16** (0.67)	0.59 (0.78)	
Sensegiving Potential	5.72* (2.56)			
Sensemaking	2.62** (0.85)		1.88* (0.93)	
Sensemaking Potential		4.02* (1.97)		
Mutual Understanding				6.56** (2.69)
N	39	39	39	13
Wald's[4] χ^2	10.49*	11.64*	25.97***	12.75**
McFadden R^2	0.417	0.359	0.325	0.457
Highest Variance Inflation Factor	1.08	1.21	1.72	1.42

[1]Instandardized regression coefficients are given with robust standard errors in parentheses. All significance levels are indicated as follows: *p < 0.05; **p < 0.01; ***p < 0.001.

[2] For Models 1–3, we first conducted panel-data ordinal regressions (using *xtologit* in Stata 15), with panels based on projects. Results of estimated variance components and likelihood-ratio tests indicate that *xtologit* does not improve over a standard ordered logistic regression. So, results of ordinal regressions (the *ologit* command) are reported in this table. We used the "*vce(cluster)*" option with clustering based on the project. The results are substantively the same when using either panel-data ordinal regressions (*xtologit*) or multiple regressions with the data clustered by project, and with a dummy for organization. Results for the other models are excluded due to space considerations.

[3] Model 4 is tested using ordinal regression (the *ologit* command) with the "vce(robust)" option to obtain robust standard errors (for consistency with Models 1–3). The results for a model including a dummy for organization are substantively the same, and are excluded due to space constraints.

[4] Using G*Power (http://www.gpower.hhu.de), medium effect size, and four (Models 1-3) or three (Model 4) predictors (excluding controls), statistical power is 0.77 for Models 1–3 and 0.36 for Model 4.

References

Bowman, B., Davis, G., and Wetherbe, J. 1983. "Three Stage Model of MIS Planning," *Information & Management* (6:1), pp. 11-25.

Choudhury, V., and Sabherwal, R. 2003. "Portfolios of Control in Outsourced Software Development Projects," *Information Systems Research* (14:3), pp. 291-314.

Conover, W. J. 1999. *Practical Nonparametric Statistics*, Hoboken, NJ: Wiley.

Creswell, J. W., and Clark, V. L. P. 2007. *Designing and Conducting Mixed Methods Research*, Thousand Oaks, CA: SAGE Publications.

Eisenhardt, K. M. 1989. "Building Theories from Case Study Research," *Academy of Management Review* (14:4), pp. 532-550.

Kirsch, L. J. 1996. "The Management of Complex Tasks in Organizations: Controlling the Systems Development Process," *Organization Science* (7:1), pp. 1-21.

Kirsch, L. J. 1997. "Portfolios of Control Modes and IS Project Management," *Information Systems Research* (8:3), pp. 215-239.

Kirsch, L. J. 2004. "Deploying Common Systems Globally: The Dynamics of Control," *Information Systems Research* (15:4), pp. 374-395.

Levina, N. 2005. "Collaborating on Multiparty Information Systems Development Projects: A Collective Reflection-in-Action View," *Information Systems Research* (16:2), pp. 109-130.

Nidumolu, S. R., and Subramani, M. R. 2003. "The Matrix of Control: Combining Process and Structure Approaches to Managing Software Development," *Journal of Management Information Systems* (20:3), pp. 159-196.

Orlikowski, W. J. 1991. "Integrated Information Environment or Matrix of Control: The Contradictory Implications of Information Technologies," *Accounting, Management and Information Technologies* (1:1), pp. 9-42.

Segars, A. H., and Grover, V. 1999. "Profiles of Strategic Information Systems Planning," *Information Systems Research* (10:3), pp. 199-232.

Shepherd, D. A., and Sutcliffe, K. M. 2011. "Inductive Top-Down Theorizing: A Source of New Theories of Organization," *Academy of Management Review* (36:2), pp. 361-380.

Tiwana, A., and Keil, M. 2009. "Control in Internal and Outsourced Software Projects," *Journal of Management Information Systems* (26:3), pp. 9-44.