Experience report: the social nature of agile teams

Elizabeth Whitworth
TRANSPOREON GmbH
elizabethwhitworth@gmail.com

Abstract

Agile software development is often, but not always, associated with the term 'project chemistry,' or the positive team climate that can contribute to high performance. A qualitative study involving 22 participants in agile teams sought to explore this connection, and answer the question: what aspects of agile software development are related to team cohesion?

The following is a discussion of participant experiences as seen through a socio-psychological lens. It draws from social-identity theory and socio-psychological literature to explain, not only how, but why *agile methodologies support teamwork and collective progress. Agile practices are shown to produce a socio-psychological environment of high-performance, with many of the practical benefits of agile practices being supported and mediated by social and personal concerns.*

1. Introduction

A review of software development literature reveals lack of basic research into the lived experience of individuals in agile development teams — or any other kind of software development team, for that matter. While pockets of literature contain strong references to socio-psychological issues, such as ego, well-being, control, and team conflict, the bulk of software development writing is practitioner based, concerning the practicalities of software construction, software development process management, and the hurdles of making it all work in a business context. Focusing purely on tasks, roles, and measurable behavioral outcomes such as productivity, however, excludes a deeper understanding of socio-psychological phenomena, such as motivation, that are commonly associated with the agile team environment.

The exploratory master's study [1] on which this report is based was conducted in the hopes of filling this gap in our understanding of software development teamwork. The focus of the study was the subjective experience of communicating and collaborating in software development teams. The major questions posed were: Why, from a socio-psychological perspective, are agile methods so strongly related cohesive teamwork? And what aspects of agile software development are related to team cohesion? Cohesive teams considered in this study were those described by participants as teams that "click," "jell," or "*really* work together" to develop software.

The study results support the view that positive psychological aspects of agile methods, such as pride, can be strongly related to the success and effectiveness of agile teams. For the original unbiased research results, please refer to the original thesis [1]. This report distills some of the findings into a more accessible format for practitioners, and has most definitely been tempered with my own team experiences in the year since the study took place.

2. Method in brief

The qualitative study was comprised of three parts: a literature review of agile, socio-psychological, and team theory; participation in the agile community; and semi-structured interviews with 22 participants who were members of agile teams. The method of analysis used was *grounded theory*. Grounded theory is not a theory itself, but a procedure for the systematic collection and analysis of qualitative data. It involves progressive iterations of data gathering and theory development, with the goal being the construction of a theory that is grounded and confirmed by raw data.

The interviews involved open-ended questions. For example: describe the last project that you were on that you felt really excited about; how was the project different from other projects you have been on?; and which agile practices did you find especially valuable for that team?

Interviews were transcribed and coded line by line to isolate theoretical concepts from the raw data. Codes were then grouped into categories of experience, with a

978-0-7695-3321-6/08 $25.00 © 2008 IEEE
DOI 10.1109/Agile.2008.53

focus on experiences or themes that held true across a number of different participants.

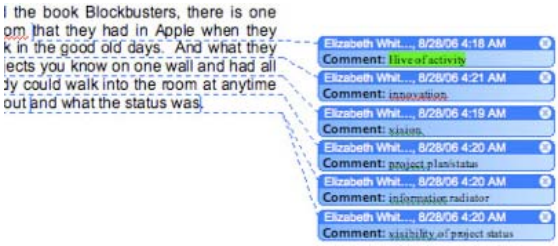

Figure 1. Grounded theory coding

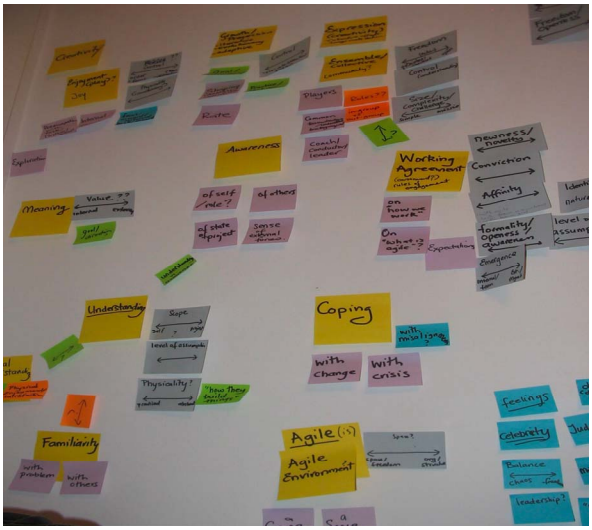

Figure 2. Category development

Further sets of interviews were then conducted to explore and confirm the developing theory against new raw data. At the end of the study, results were sent to participants and other agile practitioners to review for relevance and validity.

Extreme programming (XP) methodology was used as a basis of understanding, but the study also included teams that used Scrum or a subset of agile practices. Participants included developers, interaction designers, project managers, agile coaches, quality assurance, and documentation personnel working in a wide range of domains and organizations. The following sections discuss the results of the study, as well as my own inferences surrounding the results. Direct quotes from study participants are used to illustrate findings, and are indicated by italic text and quotation marks.

3. Agile culture: speed and ease

One of the major factors participants noted as related to team cohesiveness was the ability of team members to interact with speed and ease. This speed and ease was further related to the application and adherence to aspects of agile culture, including:

Whole team involvement. Everyone involved on the project is available, if not all the time, then at least at regular intervals. All team members are available for questions and input from the rest of the team.

Agile values. Adherence to values such as trust, openness and respect during team interactions.

Culture of action and change. Cohesive agile teams in this study were more likely to place a high value on action, initiative, and continuous improvement. Participants in such teams tended to talk in terms of action to be taken in the near future to compliment their current ideas or thoughts, and discussed their team activity as being overwhelmingly focused on progressive, actionable steps.

Collective thinking. As opposed to individual or task-based thinking, collective thinking involves whole team consideration and involvement. For example, there was a tendency for individuals in cohesive teams to say 'we' rather than 'I.' They tended to think in terms of systemic or holistic visions for the product or process, rather than in terms of individual tasks or roles.

Interestingly, these cultural aspects of agile teams were seen to exist, seemingly regardless of the specific agile practices used by the team.

4. Social identity theory

Examination of the collective culture that agile practices encourage shows many connections to the well-established *social identity theory* [2]. Social identity theory posits that individuals have several social identities corresponding to perceived membership in social groups. The salience of a given social identity depends on social context; meaning that certain contexts increase the prevalence of a given social identity as opposed to other aspects of identity, such as personal identity. Social identity is not something superficial—a number of social identities exist as part of a person's self-concept, and differing social contexts will cause a person to think and feel, as well as act differently.

Agile culture and practices were seen to heighten the presence, value, and importance of a shared project team identity, as opposed to individual or role-based identities. For example, constant immersion and engagement with the rest of an agile team, as well as the development of rituals surrounding team activity, can both encourage individuals to feel closer to the project group than to other groups in an organization.

A related concept is *in-group out-group bias*, where mere categorization into groups will instantiate a bias

that can effect individual motivation to communicate and collaborate in an organization [3]. Agile methodologies provide much more than mere categorization, however. The complex system of values, principles, and practices surrounding the term 'agile comprises a rich culture for software development. Identification with agile teams can be especially strong when compared to identification with most non-agile teams, and participants correspondingly reported increased feelings of belonging, security, comfort, and willingness to cooperate in agile teams.

Being part of an agile project team was seen to reduce the stress associated with inherent role conflicts in the software development process. For example, conflicts stemming from the developer's focus on code and technical feasibility, the product owner's focus on feature development, and the user experience professional's focus on design or usability can result in no small amount of personal tension and anxiety.

Social identity theory is a valuable tool that can aid in understanding and promoting cohesive teamwork within an organization. For example, it suggests the importance of observing and dealing with differences between agile culture and organizational culture, as such differences are likely to interfere with group identification and reduce the benefits gained from a shared culture and in-group categorization.

5. A clear objective

"Have you ever been on a really dysfunctional team?"

"In some ways that spec driven team was a bit dysfunctional, but that might have just been my perceptions. Its bits where there's more push to meet the spec than to meet the customer needs. I mean, in my view that's dysfunctional, but in terms of most software engineering teams, it's not."

One of the foundations of collective effort in agile teams was seen to be a simple and clear objective. Namely, the push for 'the most business value to the customer in the least amount of time.' Cohesive agile teams in this study were also seen to maintain a strong focus on developing *quality* software code—quality above and beyond the needs of the customer. The goals of business value and software quality have an advantage over more complex specifications-based goals, because they are goals that are easy for team members to see the value of and agree on together. They are also goals that are easy to comprehend cognitively. In comparison, a complex list of specifications is very hard for an individual to hold in their mind, identify with, and be motivated to work towards. Agile measurement of progress towards goals as a team, also allows individuals to take pride in collective efforts and membership in the project team.

6. Agile planning

The agile planning game provides a collectively defined and flexible plan for software development that will take place in the near future.

"There is a lot of tension…when you develop a feature there is only so much you can do. The product specialist will often come in with a list with 1000 items [and] there is this dance that takes place and the developer estimates well I can do 50 of those. [With agile] we will have discussions about that a lot earlier"

Visible capacity and constraints. A major effect associated with agile planning in this study is that it helps make inherent role conflicts easier to digest. For example, the practice of using story cards placed on a storyboard to define the current iteration. There is limited space on the board. If a product owner wants to add a new feature, it's clear that another feature must be shifted to off into the next iteration. Planning activities are no longer subjective, and the need for trade-offs is made visible and understandable to all parties.

Flexible and personalized planning. The agile plan is constantly adjusted to account for changing business needs. This is not a one-way street, however. Cohesive teams in this study were also seen to suit their iteration plans to the unique requirements of the project team. For example, if there is a new developer, or if a team member is on vacation or sick, the agile plan is also adjusted accordingly. The responsiveness of the agile plan to the needs and capabilities of the team was related to high levels of motivation and excitement in the team environment. Allowing developers to pick and estimate their own stories, for example, allows the synchronization of individual pace and planned team pace, and was associated with the feeling of rhythm, or flow often discussed in relation to agile teams. The added bonus is that the team can always successfully meet the goals it sets itself.

The clear goals, immediate feedback, and well-matched challenge to skill ratio for tasks assigned to individuals in the agile environment, corresponds strongly with the psychological concept of *flow* [4]. Flow is the "subjective experience of engaging just-manageable challenges by tackling a series of goals, continuously processing feedback about progress, and adjusting action based on this feedback" [5, p. 89]. Agile practices can therefore be seen to support higher levels of engagement in project activity than methods

that provide less task feedback and lower challenge-skill suitability.

Short vs. long term planning. Flexible, short term planning, such as for a week or a month, was seen to increase the perception of the current situation as temporary or non-fatal. The result is more relaxed team relations, and a significant reduction in the stress associated with planning activities. In long term planning situations, team members are investing in a plan that they must live by for the life of a project. There is therefore a strong push by each individual to get all of their needs addressed in the plan. Short term planning, on the other hand, seems to reverse this tendency. Team members become more willing to compromise or make concessions. They can live with a compromise—for the current iteration, at least—and wait until the next round for another chance to have their needs addressed.

A large planning scope also increases cognitive load, and reduces the ability of individuals to maintain an awareness of the project as a whole. The result on individuals is increased feelings of insecurity and lack of control. In contrast, individuals in an agile environment are highly aware of what everyone else is doing or planning to do, and are more likely to feel secure and comfortable in knowing that there is nothing going on that are not aware of. Overall, short term planning in a shared team environment, while still challenging, was seen to decrease both cognitive load and the perceived significance of planning activities, allowing for more relaxed team interactions, and increased security for individuals working on their tasks.

7. Regular iterative delivery

"You are working on something and at the end of the week it goes out to the customer and you get feedback right away. That's great because your work matters; every day matters."

Sense of immediacy. The delivery of working software on a regular basis was seen to result in an increased sense of immediacy of project tasks, increased energy in the team environment, and an increased willingness put aside or work through personal differences and act together to create working software. In fact, regular iterative delivery, and the resulting feedback and support from users or customers, was often mentioned by participants as *the* main motivator related to agile software development.

History of success. Consistent iterative delivery was related to high levels of trust and security in agile teams. Participants in teams able to consistently deliver software showed high levels of confidence and comfort regarding the software development effort:

"The main point is that I feel more secure about what I'm doing. Very strongly. You really prove after two weeks that this thing is really working, in a more or less stable way."

"We will do it. We always do it."

This can be held in contrast to the caution and insecurity participants described when discussing activity in teams that did not deliver on a regular basis.

Process improvement. Regular iterative delivery provides teams with the opportunity to understand and improve their process for developing working software together. Regular team retrospectives, where teams meet at the end of an iteration or release and discuss what went right, what went wrong, and to plan what will be done to improve in the next iteration, were seen to not only improve a teams process, but to have an extremely positive impact on team relations.

Retrospectives provide an important forum where team members can be heard. Meeting regularly, such as once every week or month, allows regular discussion regarding development processes, but also regarding non-task items, such as relational issues or problems with the team room environment that might otherwise never be discussed or acted upon. The mere act of getting together to discuss such team issues can increase positive feelings such as acceptance and belonging in the team environment, and was associated in this study with increased feelings of ownership and involvement regarding team processes.

It's worth noting that participants stressed the value of having a good mediator or facilitator at these meetings, which could sometimes get quite intense. Participants also noted that such meetings go a lot smoother when working in a team of people who are friendly and open-minded.

8. Whole team awareness and feedback

This research highlighted a subtle requirement for cohesive teams that is supported by agile methods, that is—constant feedback to individuals that all team members share an awareness of team activity and commitment to team goals. Awareness and feedback from the team as a whole, as opposed to through individual or partial team communication is particularly important. Sharing knowledge and receiving feedback on a team-wide basis allows a sense of 'common knowledge' that can then be used as a basis for action approved by the whole team. The two agile practices most strongly supportive of whole team awareness and feedback are daily meetings and information radiators, which will be discussed below.

8.1. Daily meetings

The daily stand-up, where the entire team meets for 10 minutes every morning to discuss progress and problems, strongly encourages information flow and problem resolution. This research highlighted a number of additional social effects associated with such meetings.

Forum for non-task based items. Participants reported feeling more comfortable bringing up non-task based issues, such as ideas for team improvement, positive feedback, or non-functional requirements, in the group forum. Some individuals mentioned that if they had an idea or piece of information related to the entire team, they felt much more comfortable bringing these issues up in a team meeting, as opposed to mentioning it to only a few of their colleagues or taking the issue directly to a project manager. Regular group meetings can therefore be seen as a forum for issues or items that might otherwise be lost in the cracks; just because people are less likely to discuss team-based issues outside of the team forum.

Social pressure and accountability. Group meetings also provide a strong amount of social accountability and pressure. Participants reported a strong aversion to attending daily meetings without having contributed to collective goals:

"That daily meeting where you kind of affirm that everybody is on the same team and you have to tell people what you are doing for the good of the project—like you can't just sit there and say, 'Well, actually I have just been working on my own thing." <laughs> "Or it's a little harder to do that right, like 'Well I haven't been working at the project at all!'"

Repercussions for bad quality or incomplete work in an agile environment tend to be social rather than punitive. For example, from questioning or joking surrounding individual action (or lack of action) in front of the group, from clear expressions of need in front of the group, or from individual expectations of disappointment or disapproval from team members.

Recognition and support. On the other hand, problems faced by individuals in completing their tasks are also explained to the group, meaning that reasonable failure to achieve goals is more likely to be met with understanding and support than in an environment where the team meets less regularly. Daily meetings also increase the likelihood that individuals will be given recognition for hard work on a regular basis. So while the agile environment can be seen as more demanding day-to-day in terms of constant reporting and follow-up, there is also a higher level of individual support and acknowledgment.

Daily meetings can also have a mediating and balancing effect on daily project activity. For example, unreasonable requests from one individual to another can be quickly corrected by others in the team, while reasonable requests are more likely to be followed up on due to expectations and reminders coming from the entire team.

Plausible, achievable chunks of activity. The value of team meetings as a forum for social accountability and support relies quite heavily on the chunking of activities into user stories. User stories provide plausible, achievable chunks of activity, which allow the group to see noticeable measures of progress.

For example, if a team member has a task that takes him the entire week to complete, then he has nothing to report at the daily meeting. He is likely to feel disengaged from project activity; he may stop listening at the daily stand-up, or may feel like he's not really part of the team. He is also not receiving a high level of feedback and support in his task. Splitting the task into small chunks, however, or allowing him to work with someone else to achieve it, would increase the speed with which he can get results, receive feedback from the team, and really feel supported and part of the project activity. User stories are therefore not only valuable from a practical perspective, but also influential to individual motivation and effort.

Regular vs. need-based communication. Regular discussions surrounding project work were also related to more positive interactions between individuals. In contrast, need-only communication was seen to set the stage for conflict-based relationships:

"Before [agile was adopted] we would meet with the product specialist once a week and they would tell us what they wanted and what was going wrong and bugs remaining, and so you get the overall impression they were demanding; they were always complaining and never satisfied……Whereas now, with our culture of communication, you meet, and yeah you still get the same kind of information…But you get the underlying feeling that they love the product and that just remains as the bedrock. And you know on the top we can deal with little bits here and there, the individual features."

8.2. Information radiators

Information radiators are tools that allow people to organize themselves as a team and measure progress towards a common goal. For example, burn down charts showing projected team progress vs. actual progress. Such measures of progress increase awareness and certainty regarding project activity, and can be associated with increased identification with the

project team and willingness to invest effort in the team endeavor.

Noticeable measures of progress. An interesting finding regarding information radiators was the effect they can have when used to regularly demonstrate noticeable measures of progress. For example, when teams publically tally the number of tests created by each developer, or physically mark story cards to indicate the completion of a story:

"And that little dot is like an endorphin rush. It's like, "Woohoo!" And the team sees the progress, and it's great."

Such indicators of progress were highly related to feelings of team cohesion, with individuals reporting an increased motivation to perform associated with such activities:

"The ritual involved in, having written a test, standing up and going over to the whiteboard, and looking around that…everyone sees what you are doing. And making your number of strokes and getting praise for it from the rest of the team, and everyone else feeling now that they, have to keep up writing the tests. So that they don't fall behind."

Information radiators can be used to increase the sense of security and commitment in the team environment. For example, a given agile team may complete 3-4 of their stories a day. When an individual gets up in front of the team to complete a ritual related to story completion, such as ringing a bell, or adding a sticker to a card, she is committing her effort to the team in a very visible way. Information radiators therefore provide a high level of awareness of individual contributions to team goals, which acts to affirm positive team sentiment towards project work. The result is a high level of confidence from individuals they are not the only ones working so hard to get the software out the door.

9. A socio-psychological climate of high performance

While this study was not focused on measures of performance, it is worth noting that the agile methods can be associated with a socio-psychological climate of high performance. Agile practices are uniquely suited to helping people deal with the high levels of complexity and uncertainty in modern-day software development environments. From a basic perceptual level, rapid feedback cycles in turn what in traditional methodologies is an overwhelmingly invisible and long-term process, into a series of tangible, manageable, and confirmatory interactions. Agile practices such as the negotiation of a flexible plan, regular iterative delivery, and group forums for whole team awareness and feedback, have been shown to be associated with high levels of awareness, security and control. The resulting team perceptions can be related to the psychological concept of *perceived self-efficacy*. Bandura defines perceived self-efficacy as "people's beliefs about their capabilities to produce designated levels of performance that exercise influence over events that affect their lives," and goes on to say that "Self-efficacy beliefs determine how people feel, think, motivate themselves and behave [6]." High levels of efficacy and control, for example, have been related to increased performance due to motivational effects [7] and reduced stress resulting in a decrease in performance errors [8].

10. Pitfalls associated with agile methods

While this research focused on the positive effects of agile methodologies, data also revealed a number of pitfalls and challenges associated with agile software development. The capacity for agile practices to support individual satisfaction and cohesive teamwork was seen to be highly dependent on how the practices were implemented and in what context, as well as on the individuals within each team.

The negative effects associated with agile methods in this study include: the tendency for some individuals to feel stressed or exhausted after spending the whole day being socially active; the capacity of agile methods to reduce the time before 'burnout,' both from increased contact with the same team member's everyday, and increased immersion in the same project activities; the inability for certain individuals or personality styles to properly integrate into agile teams; the difficulty and stress faced by individuals when transitioning into or out of the unique culture of an agile team; and road-blocks or tensions faced when trying to instantiate agile in inherently non-agile contexts, particularly the damaging tendency for individuals to engage in 'agile idealism,' even in environments ill suited to agile interactions.

Other phenomena amplified by the unique and relatively enclosed nature of agile teams included the tendency for agile teams to become overly differentiated or isolated from the rest of an organization, and the tendency for agile teams to become overly homogeneous over time. A related factor involved a propensity for agile teams to become overly attached to rituals or artifacts that had lived past their usefulness. Finally, data in this study also revealed a strong inclination for teams to revolve solely around developer activity. It was seen as more likely that agile team processes would ignore, or even increase the difficulties faced by other roles, such as

quality assurance testers, business analysts, or technical writers. Business analysts or 'customers,' for example, were seen to suffer from increased workload and responsibility, while quality assurance specialists and user interaction specialists were seen to be under-involved and/or under-appreciated in some agile teams.

Agile methods, while often successful in supporting team cohesion within a limited time span and context of application, were seen to face issues when applied over time and in differing organizational or social contexts. A number of the pitfalls discussed above illustrate some negative consequences of social identity in effect. For example, the tendency for teams to grow attached to useless rituals or become overly differentiated from the rest of the organization. While this research has focused on the positive effects of agile methods, the flip-side of the story would describe how social forces can exert a negative influence over the lives of individuals working in teams [e.g. 9].

The potential negative effects of social forces in agile teams highlights the importance of reflective practices such as retrospectives, where teams can step away from their daily work and critically evaluate their activity in relation to over-riding goals.

11. Conclusion

Team-based software development is an unstructured, complex, creative, and social, problem solving and design activity [10]. Software development efforts are further dependent on the resolution of interdependencies between team members, the synergy resulting from team wide discussion and collaboration, and the ability of all team members to share a common vision for the software to be developed. Agile methods can be seen to support, or perhaps even require the development of a collective culture over time. They support confidence, motivation, and engagement in the collective project endeavor. Erickson and Kellogg [11] note the significance of the fact that physical communication takes place in a social context, with individuals awash in ever-present social information that continuously influences numerous daily actions and decisions. Thus the practical and physical benefits provided by agile methods can be seen as constantly mediated by the social climate fostered in each team.

I hope that this report was able to draw attention to personal and social issues that, while given only lip service in traditional management, organizational, and software engineering literature, often have a large effect on the lives of software development team members. Awareness of social forces, such as social identity and in-group out-group bias, as well as individual perceptions of security, efficacy, and control, is a valuable step towards the creation and maintenance of positive and cohesive team activity in a variety of contexts; towards supporting holistic involvement of individuals in project activity, and towards reducing the stress and conflict typically associated with software development efforts.

12. References

[1] E. Whitworth, "Agile experience: communication and collaboration in agile software development teams," *M.S. thesis*, Carleton University, ON, Ottawa, 2006. http://ewhitworth.com/documents/Thesis_final.pdf .

[2] H. Tajfel, M. Billig, R. Bundy and C. Flament, "Social categorization and intergroup behaviour," *European Journal of Social Psychology*, vol. 1, pp. 49–178, 1971.

[3] P. S. Goodman and F. Olivera, "Knowledge sharing via computer-assisted systems in international corporations" *Working Paper*. Pittsburgh, PA: Carnegie Bosch Institute, pp. 98–17, 1998.

[4] M. Csikszentmihalyi, "The contribution of flow to positive psychology: Scientific essays in honor of Martin E. P. Seligman," in *The science of optimism and hope*, J. E. Gillham, Ed. Philadelphia, PA: Templeton Foundation Press, 2000, pp. 387–395.

[5] J. Nakamura and M. Csikszentmihaly, "The construction of meaning through vital engagement," in *Flourishing: Positive psychology and the life well-lived*, C. L. M. Keyes and J. Haidt, Eds. Washington D.C: American Psychological Association, 2003.

[6] A. Bandura, "Self-efficacy," in *Encyclopedia of human behavior*, V. S. Ramachaudran, Ed. New York, NY: Academic Press, vol. 4, pp. 71–81, 1994.

[7] R. A. Baron and D. Byrne, *Social Psychology,* 9th edition. Needham Heights, MA: Allyn & Bacon, 2000.

[8] I.-C. Huang, "Self-esteem, reaction to uncertainty, and physician practiced variation," *Social Behavior and Personality*, vol. 26(2), pp. 181–194, 1998.

[9] J.R. Barker, "Tightening the iron cage: Concertive control in self-managing teams,"*Administrative Science Quarterly,* vol. 38 (3), pp. 408–437, 1993.

[10] A. Warr, and E. O'Neill, E., "Understanding design as a social creative process," in *Proceedings of the 5th Conference on Creativity & Cognition* (C&C '05), 2005, pp. 118–27.

[11] T. Erickson and W.A. Kellogg, "Social translucence: An approach to designing systems that support social processes," *ACM Transactions on Computer-Human Interaction*, vol. 7, pp. 59–83, 2000.