

The Social Nature of Agile Teams

Elizabeth Whitworth and Robert Biddle

Human-Oriented Technology Lab

Carleton University

Ottawa, Canada

elizabethwhitworth@gmail.com robert_biddle@carleton.ca

Abstract

Agile methodologies represent a ‘people’ centered approach to delivering software. This paper investigates the social processes that contribute to their success. Qualitative grounded theory was used to explore socio-psychological experiences in agile teams, where agile teams were viewed as complex adaptive socio-technical systems. Advances in systems theory suggest that human agency changes the nature of a system and how it should be studied. In particular, end-goals and positive sources of motivation, such as pride, become important. Research included the questions: How do agile practices structure and mediate the experience of individuals developing software? And in particular, how do agile practices mediate the interaction between individuals and the team as a whole? Results support an understanding of how social identity and collective effort are supported by agile methods.

1. Introduction

In the 2nd edition of *Extreme Programming: Embrace Change* [1], Kent Beck added to his original work a new first sentence – “Agile is about social change.” Agile or ‘lean’ methodologies offer an alternative to traditional software engineering processes, much in the way that lean manufacturing [2] was offered in place of the assembly line – with significant human and social gains underlying increases in business value [3][4]. While the origins of the technical and engineering aspects of agile go back some time [5], agile methods remain distinctive in that they explicitly integrate behavioral and social concerns into software development methodology. Indeed, the ‘people’ focus of agile methods is singled out as an essential factor in their success and growing popularity [6][7].

The purpose of this research was to gain insight into the non-technical workings of agile teams. While agile practices support and motivate social activity during software development [8], there is still

a limited understanding of how social forces come to play in project teams. Thus we conducted an exploratory grounded theory study, with a focus on socio-psychological experiences in agile team environments.

1. Theoretical Framework

The framework for this study defines agile software development teams as complex adaptive socio-technical systems. Traditional general systems theory [9][10], however, seemed inadequate to support research into the socio-psychological phenomena occurring in agile teams. In what has been called the ‘first paradigm shift’ in systems thinking [11], systems can be organized into three types: deterministic, or mindless systems; animated, or unminded systems; and social, or multiminded systems [11][12]. These types of systems are based in the physical sciences, the natural sciences, and the social sciences respectively, with the development of general systems theory historically grounded in the first two fields of study.

The introduction of human agents into a system adds additional complexity in that each individual within a *social* system is purposeful, having his or her own free will or intent. This development of systems theory suggests a number of things to the study of human activity in teams.

Firstly, it outlines the importance of observing *system and subsystem goals*, and how they align with each other, as opposed to adhering to traditional cause and effect models of human behavior. This teleological view, concerning output or end state can be held in contrast to a deterministic view, which concerns itself with only the effect of input into the system [12]. Such analysis highlights the importance of studying positive psychological phenomena, such as pride, that have been understudied in psychology [13], and have been given particularly little attention in management theory and practice [14].

Secondly, it suggests observation of *interactions between system levels*. Merali [15] notes that

boundaries separating a system from its environment, which play an important role in traditional systems theory, are inherently different for social systems; primarily due to the dynamic nature of social network interactions and the permeable nature of organizational boundaries in practice. The ‘non-modular’ nature of humanity is an important point of study, since process and system models that do not take this into account work very differently in practice than in theory.

Thus rather than focus on either the team as a whole, or the individuals in the team, this research sought to explore the interaction between them. The distinction between the group as a collective entity and the individuals within it follows in the tradition of psychoanalyst and group therapist Wilfred Bion [16]. Bion held that a group exerts considerable influence on each of its members. The following study therefore involved the conceptualization of agile teams as comprised of individuals, the group as a whole, agile practices, and the socio-psychological relationships between them.

This paper focuses on the research questions: How do agile practices structure and mediate the experience of individuals developing software? In particular, how do agile practices mediate the interaction between individuals and the team as a whole?

2. Method – Grounded Theory

Grounded theory [17][18][19] is a research methodology that provides a set of procedures for the systematic collection and analysis of qualitative data. Grounded theory is characterized by use of the constant comparative method of analysis. Data and abstract concepts are constantly compared to each other, ensuring the development of an integrative theory that is firmly and empirically grounded in raw data.

Twenty-two participants were recruited through networking with members of the agile software development community. Participant roles included five agile developers, three interaction designers, one user experience manager, five project managers, two agile coaches, two quality assurance specialists, one technical documentation specialist, and three developers who had previously been project managers. All but two of the participants had previously worked in non-agile teams. Participant teams ranged from teams that utilized most or all agile practices, to those that only utilized a few agile practices and could be characterized on the whole as ‘non-agile.’

There were sixteen male participants, and six female participants. Participant interviews investigated the subjective experiences of individuals in agile software development teams. Semi-structured interviews were chosen in order to maintain focus on the theoretical framework, while still leaving room for phenomena significant to participants to emerge.

Each interview was audio recorded and transcribed. The transcriptions were broken down into discrete parts and incidents were identified, conceptualized, and named in the process of open coding [19]. Open coding was conducted line by line to ensure thorough grounding and critical thinking about data. Axial coding was then used to examine the relationships between data. An emphasis was put on developing valuable analytical categories to explain phenomena, as opposed to a tightly woven theory (as discussed by Charmaz [17]).

Glaser [20] offers two major criteria for an emerging theory: that it fits the situation, and that it helps people in the situation make sense of, and manage their experience better. The first criteria was ensured by ‘purposeful grounding’ of theory in interview data [19]. Abstract concepts were tested against data, and only validated concepts appear in final study results. Secondly, feedback elicited from participants regarding study results suggests that findings are accessible, relevant, and useful to software development practitioners, in accordance with requirements for credibility in grounded theory studies [18].

3. Results

In our study, we asked about software development teams characterized by strong feelings of excitement, and participants discussed well functioning teams that ‘clicked,’ ‘gelled,’ or ‘*really* worked together’ to successfully develop software. Such ‘cohesive’ teams can be distinguished from non-cohesive teams, which were not associated with feelings of excitement. This is separate from the distinction between agile and non-agile teams. Our analysis therefore included the question: what characteristics of agile teams are related to team cohesion?

In the sections that follow, we explain what emerged from our study. We quote from our interviews, identifying the selection by three radices, e.g. (L.4.35), specifying the interview batch, the number of the quoted participant within each batch, and the interview paragraph.

3.1 Whole Team Awareness and Acceptance

This research highlighted a subtle requirement for cohesive teams that is supported by agile environments; that is, constant feedback to the individual that all team members share awareness of team activity and commitment to team goals. Awareness and feedback elicited from the team *as a whole*, as opposed to through individual or partial team communications, was found to be particularly important. Sharing knowledge and receiving feedback on a team-wide basis was seen to allow a sense of ‘common knowledge,’ which could then be used as a basis for action that has been ‘approved’ by the whole team:

And its basically having the sort of collective thing...agreement that there is a problem and then agreement that there is just a way we can improve things rather than having a...whinge and then not being able to do; or feeling powerless in terms of – oh we have to live in this environment.

(L.4.13)

Daily or weekly team meetings, in particular, were shown to be an important motivator of both individual and group action. Such group forums were related to the ability of individuals to engage in initiatives that affected the whole team. Shared team meetings and environments, therefore, were seen as forums where team members could bring up non-task based items that were of interest to the whole team, such as software architecture, non-functional requirements, and user or customer feedback that might otherwise be ‘lost in the gaps’ of individual or non-team-wide communication.

Agile participants expressed a strong aversion to showing up to such meetings without having contributed to collective goals:

And I think [it's important to have] 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 project – like you can't 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!' <laughs>.

(T.4.39)

Repercussions for bad quality or incomplete work in cohesive teams in this study were highly social rather than punitive. Social accountability was seen to come from frequent questioning or joking

surrounding individual action (or lack of action), from a clear expression of need in front of others in the team, or from individual expectations of disappointment or disapproval from the rest of the group. Conversely, problems faced when attempting to complete a task were also explained in the team forum, meaning that reasonable failure to achieve goals was met with understanding and support from the rest of the team. Such whole team awareness and acceptance as supported by agile methods was associated with an increased sense of individual responsibility and self-worth regarding project activity.

Several aspects of the agile environment were seen to amplify the effectiveness of team meetings as a forum for motivation, accountability, and action. In particular, the limitation of team expectations to plausible, achievable, and noticeable chunks, such as stories or iterations, was found to be important.

Cohesive agile teams would often assign to individuals relatively limited responsibilities that were agreed upon by the entire team. This would occur for non-task-based items as well as regular software development tasks. After a short period, when things were still fresh in people’s minds, teams would then revisit these items, and observe the progress made towards them or towards over-reaching goals:

And that's where the positive thing comes out of it because what happens is you might decide to prioritize the ownership of these actions that you are going to take to further mitigate or improve the bad situations. And so somebody has got to have responsibility for maybe just 2 out of 10 of them...and what happens is that the next stretch is like somebody will go over and say 'Well this, this and this was dealt with, is everybody satisfied that it was dealt with?'

(L.4.11)

By assigning small chunks of action to individuals, and by maintaining *regular* team awareness of member activity, agile teams allow for an extremely high level of social support and accountability during software development. Team-wide sharing and discussion in agile teams was seen to support feelings of satisfaction, acceptance, and belonging; particularly when compared to development environments where team member interactions were relatively infrequent or on a need-only basis. Reasonable failure to complete a task in teams where individual activity and issues were not regularly shared was associated with exasperation, annoyance, and stress in the team environment.

Need-only communication between team roles was further related to conflict-based relationships and feelings of dissatisfaction towards group work:

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 was like they were demanding; they were always complaining and never satisfied...

In contrast, agile team members displayed an awareness of the underlying motivations of their team members and increased receptiveness to their requests:

...whereas in you know, with our culture of communication with them, you meet, and yeah you still get the same kind of information...if things are missing or they wanted to go in this direction and not that direction. But you get the underlying feeling that they love the product and that just remains as the bedrock. And then you know on the top you know we can deal with the little bits here and there, the individual features. But I think you can get a better sense of just the communication between the various groups.

(T.2.22)

3.2 Holistic Team Involvement, Awareness, and Collective Effort

Having all project members present in team meetings or team rooms, as per agile practice, physically supports ease and speed of communication and collaboration. We also noted additional social effects. Firstly, the presence of all (or most) team members in agile environments seemed to increase perceptions of the importance of the collective endeavor:

So all of those people came everyday to the Scrums and talked and actually even the marketing person came to these Scrums and talked. I think what I liked about it was the enthusiasm that everybody had for the project. Having those people show up everyday and having people, you know, just really enthusiastic about what we are doing, what we are trying to achieve and all really backing the project.

(T.4.1)

Secondly, individuals were found to appreciate the holistic awareness provided by a shared environment. Individuals that could be assured that

there was nothing occurring elsewhere in the project that they were missing or should be aware of were seen to feel more confident in both their own work and in the work of others in the team:

Like I might be working on my own little bit of the job. But I will kind of be aware of what the guy next to me is doing as well, and what the guys across the other side of the corridors are doing. Because you are right there and you hear the discussions. You are not actually involved in the discussions but you have got an opportunity to be involved in them. If someone starts talking about something and they start talking about something that is related to what you are doing, then you can prick your ears up and you can join in the conversation if necessary. Whereas if they are on a different floor, then you are never aware that that conversation has taken place, So they might be working on something that affects you but you didn't know that. It's good to just...it's much easier to just keep your ear to the ground and get a good picture of what is going on when everyone is sitting together.

...

And you don't find that having these conversations going on around you is distracting?

Ah strangely enough no.

(L.3.22)

Participants revealed a strong interest in general indicators of agitation or excitement amongst other team members, in addition to the information needed to complete their own tasks. Participants were further concerned with detailed information, such as who was having problems, what the problems entailed, and what was being done to fix them. Awareness of team-wide problems, issues, successes, and progress, was related to a sense of comfort, sense of control, and feelings of responsibility from individuals towards the project as a whole. The absence of such awareness, on the other hand, was associated with insecurity, discomfort, and a lack of sense of control in the project environment:

I felt uncomfortable at the moment that I was further away from where things that were happening... you heard less about things already being solved, or why the build was not running at that moment.

(X.4.20)

Often, a lack of awareness of what was going on within a project was related to a lack of trust, and lack of team cohesion or ‘togetherness:’

I was recently working with a non-agile team and I couldn't believe how much I did not feel a part of the team. I didn't understand where the information was coming from – they didn't have scrums. You know, there was some flow of information but it was really hard to figure out where it was...I mean was it email that I wasn't getting? Was it a meeting that I didn't know about? Like there seemed to be this weird way of getting information that I had to try to squirrel and figure out before I could get in on the stuff. But I never really felt I was completely part of the team.

(T.4.38)

A shared development environment was seen to provide assurance to individuals that the bulk of activity occurring on their project had been validated by the group, and that there would be little need for backtracking or rehashing of work at a later date. Coordinating practices such as automated testing and continuous integration were also valuable in this regard. Team-wide awareness and control as supported by agile practices was seen to allow a focus on quality and craftsmanship in a way that was previously difficult to achieve within a team:

The team pays a lot of attention on quality so one of the things that agile is introduced here, for example, it's been automated testing. So the team has just gone nuts, I think it's probably the best way of putting it, on automated tests, and we have no other product in the building like it...it's been pretty constant over time and I think the team is justifiably pretty proud of that. And I think this really brought them together.

(T.1.8)

Agile practices were thus seen to increase the level of investment and involvement in the collective endeavor, which were further associated with high levels of helping, trust, and goodwill in the team environment:

Do you feel when you are doing your work any dependency from the team?

Yes. But that is not any different between agile and non-agile projects. You can have a situation where the team is depending on you at any time. The component is not going to work until my bit is done. So even if they finished with

their tasks yesterday and I wasn't going to get my bit done until the end of this week, then they are really depending on me to get my bit done.

It's actually I think, accentuated on a non-agile team that dependency. Because on an agile team, if my bit didn't look like being finished until the end of the week, then um, then they wouldn't be twiddling their thumbs waiting for me to finish it. They would be like “What can I do to help?”

And people still do that on non-agile teams, people are not totally siloed. But it's just like breathing on an agile team. that's just like, that's the normal thing. Whereas on a non-agile team it can be like “Well I'm helping you out here; I'm doing you a favor.” And you are like “No you are not. We are both doing the same number of hours on this. We are both aiming for the same goal. We are both just doing our jobs.

(L.3.17)

3.3 Information Radiators and Noticeable Measures of Progress

Individual awareness and certainty regarding current project activity, as well as the validity of the team endeavor, was strongly related to the use of information radiators [7]. Information radiators, such as burn charts, allow teams to clearly visualize current project status and what is required to complete goals. Such information radiators were discussed as being invaluable sources of motivation, excitement, and team cohesion:

I think basically what the burn-up charts did was giving us a very good overview of where we are standing. And giving us a sense of control over the situation. Giving us a sense of the things, where we suddenly saw that “Yes we can make it. Where there were other things that we already had a feeling that it won't work, but it we couldn't put the finger on it. And the burn chart gave us a way to actually, I don't know how to say – materialize? – this feeling. And to get some very simple way to communicate to management, to point to something and say – “We have a problem here, and we are not quite sure what to do, and please help us.” And I think that's what saved us.

Saved you from what?

From... ah, I don't know the saying in English. But in German you would say

'drowning in chaos'...I think we all had this sense of urgency, but everyone was working for himself alone, not as a team. Because we didn't have the time to organize. And the burn chart gave us an opportunity to organize around it...to make a team again. And to give it a common goal. To make the chart look like it will work...now that I think about it, before hand, everyone had his own tasks, his own stories, and he was concentrating on that and was kind of blindsided. And, um, the information radiator was a way to show that there was a team goal that we had to reach together.

(X.3.23-28)

Team awareness of individual activity as supported by information radiators was discussed as a strong motivator in the agile team environment:

I told you about the button that we have to press? Well in fact we also have a website that you can go to and press the button there and get the same impact. And I still like to get up and press the physical button. Because one thing is that it is just more fun. Cool in some kind of way. And the team notices who is pressing the button.

And even though when you press the button on the website there is some sound played, and the state of the traffic light changes. It's often something that the rest of the team won't notice as much. But when I am going there to press the button, they just look at me, and they notice the kind of team directions that's missing with the software tools...the interaction with the team, is something that is very important. I feel as part of the team. And that's happening while I walk up.

(X.3.6)

In this example, a situation where the whole group notices individual contribution is compared to a situation where the contribution is made, but is less noticeable to the rest of the team. Cohesive agile environments in this study all involved a high level of awareness of individual contribution to collective goals, usually by way of a highly visible information radiator in a shared team environment (see also [21]). In the example above, the team was using a traffic light that lit up green when a working software build was running, and red when it was broken.

Other measures of progress used in teams included the use of toys that make sounds, marks on a project board in the team room, or simply the evolution of the final software product itself; often it would be something quirky or fun. The important thing seems to be that everyone in the team notices what is happening:

And that little dot [on the project board] is like an endorphin rush. It's like, 'Woohoo!' And the team sees the progress, and it's great.

(O.3.16)

An interesting factor was the frequency of these measures of progress. Events described by participants, such as writing a unit test, or a pair integrating their code into the system, occurred a number of times every day in highly cohesive team environments. There seems to be value, therefore, in frequent signs of progress towards collective goals. Such indicators were seen to strongly support individual motivation to contribute to team efforts:

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

(X.3.2)

Part of the value of such awareness of individual actions was seen to be in its capacity to strongly affirm team sentiment regarding the project and team environment. Noticeable indications of progress seemed to allow team members to feel secure in the fact that everyone in the team was working towards a common goal rather than working at odds. Assurance that everyone was 'on board' in this way was associated with increased identification with the team, and appreciation of the team goal as something more valuable and worth investing in than if these affirmations were not present, or if there was a lack of involvement on the part of any individual in the team, for whatever reason. Regular affirmations of commitment were highly related to trust, respect, positive feeling in the team environment, and increased willingness to collaborate.

3.4 Agile Culture

Activity in the software development environment was discussed as highly mediated by team expectations. Participants in this study talked of a tangible 'agile' culture, adopted to a greater or lesser extent across ostensibly agile and non-agile teams. Interview participants who strongly identified themselves with agile software development ('agile participants') exhibited common understandings and values. These existed relatively independently of the particulars of software development methodology or practice, although agile practices were supportive of

such culture. Interviews revealed the presence of an ‘agile ideal’ regarding both individual and team activity that most agile participants were striving to live up to. Though exact interpretations of ‘agile’ are many and varied, the core of this ideal for most participants seemed based on editions of *Extreme Programming: Embrace Change*, by Kent Beck [1].

Three characteristics associated with agile culture seem particularly pertinent to this study. Firstly, agile teams were seen to encourage an environment of openness and respect. In particular, participants in agile teams showed a heightened awareness of the actions and opinions of others in their team, and a better understanding of their own opinions and roles in relation to the rest of the team. While agile practices instantiate team activity to be observable or transparent, cohesive agile teams were seen to ubiquitously encourage openness and honesty, from developers to management. It is expected in such teams that problems, issues, points of contention, and points of interest, will be brought up immediately and discussed openly. As one agile team leader noted:

This isn't a place that you go and hide
(O.2.14)

Many agile team members were seen to express feelings of liberation, and no small amount of relief at being able to develop software in such an environment. Secondly, agile participants showed a strong inclination for whole team consideration and involvement. Participants in agile teams would overwhelmingly talk about ‘we’ and ‘us’ rather than ‘I’ or ‘me,’ and would talk in terms of holistic and systemic visions of the software product or process rather than in terms of individual tasks or roles. Agile participants were seen to be particularly concerned with involvement and inclusiveness across the entire software development team. Such concerns were supported by agile practices, which allow for ease and speed of communication and collaboration, frequent interaction between team members, and high levels of immersion in project activity:

Throughout the project, whenever you think that you need to find anything out at all you just go and talk to people. That's another thing that is very different in agile, is that all roles are involved throughout the entire process.
(L.3.20)

Finally, agile participants in this study placed high value in action, initiative, and continuous improvement. Participants involved in cohesive agile teams, particularly, discussed change in a very immediate sense. They talked in a way that was much

more active, probing, and inquisitive, and tended to display a kind of exploratory idealism regarding the changes that they could make to improve their team and project environments.

Agile participants in general also showed a tendency to talk in terms of actions to be taken in the near future that would complement ideas or thoughts that came out during the course of a discussion or interview. In comparison, non-agile participants showed a lack of active participation in their working environment, and even a certain degree of submissiveness and lethargy regarding the change that was possible in their team or organization.

3.4 Pitfalls in Agile Environments

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 ‘on’ or 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 from 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 seen to be 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 agile software development teams to revolve solely around developer activity. This was seen to make it more likely that 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 within a limited time span and context of application, were therefore seen to face issues when applied over time and in differing organizational or social contexts. Further study of diverse configurations of agile practices, management styles, and team membership is required in order to understand the resultant and underlying effects on individuals and teams. Such study will help to identify the critical success factors involved in the implementation of agile methodologies, and to reduce the stress and difficulties faced by many agile teams.

4. Implications

4.1 Social Identity

While recent research regarding self-regulating work teams has centered around organizational, rather than social psychology [22], the social perspective seems essential in describing activity in agile software development teams. Social identity theory [23], in particular, explores individual psychology within a social context, and offers a number of valuable insights into agile teamwork.

In social identity theory, individuals have several ‘social identities’ corresponding to perceived membership in social groups [24]. In contrast, personal identity stems from self-knowledge of unique personality traits and interpersonal relationships. The salience of a social identity in an individual at any one time is seen to vary based on social context. Certain contexts will therefore increase the prevalence of social identity as opposed to personal identity, and vice versa. In sum, social identities are seen as an essential aspect of a person’s self-concept, and differing social contexts will cause a person to think and feel, as well as act differently.

Agile methods were seen in this study to heighten the presence, value, and importance of project team identity as opposed to individual or role-based identity. Constant immersion and engagement with the team as a whole, for example, and the development of rituals surrounding team activity, were seen to support the development and prevalence of a shared identity. Agile participants showed a

corresponding reduction in identification with software development roles; roles which are often internalized and linked to key aspects of an individuals’ self-perceptions and self-concept [25]. Increased identification with the agile team was thus related to improved communication, and reduction in the stress associated with inherent role conflict (see [26]).

An idea closely related to social identity theory is the notion of social comparison [27]. The theory of social comparison holds that positive self-concept is fundamental to psychological functioning, and is established through evaluation of self as compared to similar others. Thus positive self-image and self-esteem can be gained through comparison to other group members, and through identification with a group holding a high amount of prestige. Findings in this study were consistent with this theory. Members of cohesive agile teams noted positive feelings in association with recognition of their abilities within the team, and pride in project group membership.

While Tajfel and Turner [28] discuss how individuals and groups gain prestige through in-group out-group comparisons, however, prestige-based comparisons between agile and non-agile teams in this study were relatively rare. Teams in this study were more likely to compare their progress to the expectations and reactions of users and business towards their software product. Teams were additionally seen to compare their activity to the elusive ‘agile ideal.’ By providing focus for teams in the form of working software and optimal agile process, agile methodologies can therefore be seen to support prestige-based identification with the project group, and to mediate common social occurrences such as inter-group comparison.

4.2 The Importance of Agile Culture

Studies of social identity highlight the minimum conditions for in-group and out-group bias. Mere categorization into groups will instantiate a bias [23], which can further effect motivation to communicate and collaborate in organizations [29]. Agile methodologies were seen to provide much more than mere categorization, however, with the complex system of values, principles, and practices surrounding the term ‘agile’ comprising a rich culture for software development.

Identification with teams practicing agile was seen to be especially strong when compared to most non-agile teams, and commitment to collective goals and cohesiveness in agile team environments was seen to be equally high. While this could be due to

the agile environment itself, identification by individuals in non-agile teams with agile culture speaks to a certain appeal, or 'stickiness' [30], of the methods to members of the software development community. Thus 'agile' in this study was revealed to be more than a software development methodology. Significant aspects of cohesive teams were seen to stem from instantiation of an agile culture in teams, apart from any particular software development practice.

The link between agile culture and cohesive agile teams suggests the importance of observing the interaction between agile culture and organizational culture. Participants discussed difficulties when dealing with organizational culture or management styles that were inconsistent with agile values. Seen in light of social identity theory, identification with one group culture is likely to interfere in identification with the other. This conflict is likely to reduce benefits, such as increased willingness to collaborate with group members, that can be gained from a shared culture and in-group categorization.

4.3 Concertive Control and the Iron Cage

Agile environments were seen in this study to support identification with the project team, commitment to a collective goal, and high levels of social awareness and accountability. The positive team outcomes that stem from such environments are numerous. Heightened awareness in the team environment, for example, was related to reduction in social loafing; i.e, the tendency for individuals to expend less effort on a task when working in groups than when working alone [31].

A danger in such environments, however, is the dominance of 'concertive control' in the lives of individuals. In his article, *Tightening the Iron Cage: Concertive Control in Self Managing Teams*, James Barker [32] discusses the strong social forces present in post-bureaucratic structures such as self-regulating teams. Barker outlines a case study in which a self-regulating team suffers from normative (group) control that becomes constrictive to the individuals within it, resulting in high levels of stress and disenchantment with collective efforts. Barker describes a situation where heightened social awareness and involvement in a team eventually results in a heavy-handed system of social control, within which team-members suffer from extreme loss of freedom and reduction in job satisfaction. Thus what can be seen as the major benefit of self-regulating teams – their ability to motivate team members to work together in a flexible manner

towards collective goals – can be waylaid by over-application of social controls.

Such history regarding self-regulating teams has implications for agile software development. Initial work on self-regulating teams conducted at the Tavistock Institute of Human Relations (e.g. [33]), for example, was focused on individual well-being. Later teamworking initiatives, however, marginalized the democratic and 'Humanization of Work' (HoW) aspects of teamwork and focused instead on organizational performance measures [34]. Application of teamwork theory in current organizational contexts is now often characterized by heightened organizational and system control over individuals, resulting in environments akin to the authoritarian and bureaucratic control structures that the such teamworking methods were designed to replace [32].

Practitioners of agile methods should take care to avoid utilizing agile methods in ways that support collective control, but that do not support individual freedom and team flexibility. An aspect of agile methodologies and culture that already seems to be underemphasized in literature is the propensity for agile teams to be reflective, adaptive, and self-regulating. Agile teams are often encouraged to change and improve based on team member input, and can be seen to be thus striving for optimal alignment of employees and technology as specified by socio-technical systems theorists [35]. Cohesive agile teams in this study almost ubiquitously engaged in some level of team reflection or retrospection [36]. By stepping back and evaluating their activity, teams were seen to become aware of inefficiencies in their development efforts stemming from social or physical constraints, and to self-correct in a manner that is supportive of optimal system performance.

This study highlighted other aspects of agile teams that seemed to counter the dominance of group controls. 'Agile culture,' with its focus on delivering the most business value in the least amount of time, seemed particularly important. Agile conceptual tools such as 'do the simplest thing that will possibly work,' and 'you ain't gonna need it,' for example, were seen to mediate group tendencies towards dogmatism, and maintain team flexibility and efficiency. The action-based nature of agile culture can also be seen to reduce group effects leading to inertia in large teams.

Implementation of the technical aspects of agile methodologies without concern for associated cultural factors may leave room for the development of constrictive or unproductive team norms and activity – an 'iron cage' of agile control. A focus on productivity and performance to the exclusion of

other considerations may further result in burn-out or stress in the team environment.

4.4 Implications for Distributed Software Development

Finally, the importance of whole team awareness and feedback in agile software development teams has implications for the design of methods for distributed software development. Current tools and methods often focus on detailed technical development and person-to-person interaction. This study suggests, however, that group-based awareness and communication is also needed. Such whole team interaction was shown to be essential in fostering the motivation, involvement, and cohesiveness associated with agile teams. In particular, this study highlights the importance of tools that allow for the collection and display of normative information (see [37]), and tools that display the linkages between individual efforts and collective efforts and goals.

5. Conclusion

This study suggests that there are strong social forces at play in agile teams that underscore the value of agile methodologies. Agile practices such as daily team meetings were seen to provide high levels of social accountability and support, and awareness of activity in a project, such as provided by information radiators, was seen to increase feelings of security and control in the team environment. Exploration and understanding of socio-psychological and cultural issues in software development teams is essential as the field of software engineering moves to embrace post-bureaucratic work structures such as self-regulating teams.

6. References

1. K. Beck, with C. Andres, *Extreme programming explained: Embrace change* (2nd ed.), Addison-Wesley Professional, Reading, MA, USA, 2004.
2. J. P. Womack, D. T. Jones, and D. Roos, *The machine that changed the world: The story of lean production*, Rawson Associates, New York, NY, USA, 1990.
3. J. Highsmith, *Agile software development ecosystems*, Addison-Wesley, Boston, MA, USA, 2002.
4. M. Poppendieck, *Lean thinking: The theory behind agile software development*, Poppendieck, LLC., 2002.
5. C. Larman and V. R. Basili, "Iterative and incremental development: a brief history", *IEEE Computer*, 36(6), 2003, pp. 47-56.
6. B. Boehm and R. Turner, *Balancing agility and discipline: A guide for the perplexed*, Addison-Wesley, Reading, MA, USA, 2004.
7. A. Cockburn, *Agile software development*, Addison-Wesley, Reading, MA, USA, 2002.
8. H.M. Robinson & H. Sharp, *The social side of technical practices*, Proceedings of the Sixth International Conference on Extreme Programming and Agile Processes in Software Engineering (XP2005), 20-22 June, Springer Verlag, Sheffield, 2005.
9. L. von Bertalanffy, "An outline of general system theory", *Britisch Journal of Philosophie of Science*, 1, 1950, pp. 139-164.
10. A. Leonard, with S. Beer, *The systems perspective: Methods and models for the future*, 1994. Retrieved March 6th, 2006 from http://www.futurovenezuela.org/_curso/6-sysmeth.pdf
11. J. Gharajedaghi, *Systems thinking: Managing chaos and complexity: A platform for designing business architecture*, Butterworth-Heinemann, Boston, MA, USA, 1999.
12. R. Ackoff, *Ackoff's best, his classic writings on management*, John Wiley & Sons, New York, NY, USA, 1999.
13. C. M. Peterson and M. E. P. Seligman, "Positive organizational studies: Lessons from positive psychology". In K. S. Cameron, J. E. Dutton and R. E. Quinn (Eds.), *Positive organizational scholarship: Foundations of a new discipline*, Berrett-Koehler, San Francisco, CA, USA, 2003, pp. 14-27.
14. S. Ghoshal, "Bad management theories are destroying good management practices", *Academy of Management Learning and Education*, 4(1), 2005, pp. 75-91.
15. Y. Merali, "Complexity and information systems". In J. Mingers, & Willcocks. L. (Eds.), *Social theory and philosophy of information systems*, John Wiley & Sons Ltd, Sussex, UK, 2004, pp. 407-446.
16. W. R. Bion, *Experiences in groups*, Tavistock Publications, London, UK, 1961.
17. K. Charmaz, "Grounded theory". In J. A. Smith, R. Harr and L. Van Langenhove (Eds.), *Rethinking methods in psychology*. Sage Publications, Thousand Oaks, CA, USA, 1995.
18. B.G. Glaser and A. Strauss, *The Discovery of Grounded Theory*, Aldine, Chicago, IL, USA, 1967.
19. A. Strauss and J. Corbin, *Basics of Qualitative Research*, Sage, Newbury Park, CA, USA, 1990.
20. B.G. Glaser, *Basics of grounded theory analysis: Emergence vs. forcing*, Sociology Press, Mill Valley, CA, USA, 1992.
21. H. Sharp, H. Robinson, J. Segal, & D. Furniss, *The Role of Story Cards and the Wall in XP teams: A Distributed Cognition Perspective*, Proceedings of the Agile 2006 Conference, 23-28 July, Minneapolis, Minnesota, IEEE Computer Society Press, 2006, pp. 65-75.
22. D. R. Ilgen, J. R. Hollenbeck, M. Johnson, and D. Jundt, "Teams in Organizations: From input-process-

- output models to IMOI models”, *Annual Review of Psychology*, 56, 2005, pp. 517-543.
23. H. Tajfel and J.C. Turner, “The social identity theory of inter-group behavior”. In S. Worchel, and L.W. Austin (eds.), *Psychology of Intergroup Relations*, Nelson-Hall, Chicago, IL, USA, 1986.
 24. M. A. Hogg and G.M. Vaughan, *Social Psychology* (3rd ed.), Prentice Hall, London, UK, 2002.
 25. R. A. Baron and D. Byrne, *Social Psychology* (9th ed.), Allyn & Bacon, Needham Heights, MA, 2000.
 26. K. D. Williams and S. J. Karau, “Social loafing and social compensation: The effects of expectations of coworker performance”, *Journal of Personality and Social Psychology*, 61, 1991, pp. 570-581.
 27. L. Festinger, “A theory of social comparison processes”, *Human Relations*, 7, 1954, pp. 117-40.
 28. H. Tajfel and J. C. Turner, “An integrative theory of intergroup conflict”. In W. G. Austin and S. Worchel (Eds.), *The Social Psychology of intergroup relations*, Brooks/Cole, Monterey, CA, USA, 1979.
 29. P. S. Goodman and F. Olivera, “Knowledge sharing via computer-assisted systems in international corporations”, *Working Paper 98-17*, Carnegie Bosch Institute, Pittsburgh, PA, USA, 1998.
 30. M. Gladwell, *The Tipping Point: How Little Things Can Make A Big Difference*. Little, Brown, and Company, New York, NY, USA, 2000.
 31. B. Latane, K. Williams, and S. Harkins, “Many hands make light work: The causes and consequences of social loafing”, *Journal of Personality and Social Psychology*, 37, 1979, pp. 822-832.
 32. J. R. Barker, “Tightening the Iron Cage: Concertive control in self-managing teams”, *Administrative Science Quarterly*, 38(3), 1993, pp. 408-437.
 33. E. L. Trist and K. Bamforth, “Some social and psychological consequences of the longwall method of coal getting”, *Human Relations*, 4(1), 1951, pp. 3-38.
 34. F. Mueller, S. Proctor, and D. Buchanan, “Teamworking in its context(s): Antecedents, nature and dimensions”, *Human Relations*, 53, 2000, pp. 1387-1424.
 35. F. E. Emery and E.L. Trist, ‘Socio-technical systems’. In F.E. Emery (ed), *Systems Thinking*, Penguin, Harmondsworth, UK, 1969, pp. 281-296.
 36. N.L. Kerth, *Project retrospectives: A handbook for team reviews* (2nd ed.). Dorset House Publishing Company, New York, NY, USA, 2001.
 37. B. Whitworth, R. B. Gallupe, and R. McQueen, “Generating agreement in computer-mediated groups”, *Small Group Research*, 32(5), 2001, pp. 625-665.