Please cite this paper as:

Stacey, P.K. (2014) "Its Time to Act: Understanding and Assessing Agility in Information Systems Development", Lancaster University Management School, Working Paper series, Working Paper 2014:7

Lancaster University Management School
Working Paper 2014:7

Its Time to Act: Understanding and Assessing Agility in Information Systems Development

Dr. P.K. Stacey

The Department of Management Science
Lancaster University Management School
Lancaster LA1 4YX
UK

Abstract

This paper focuses on addressing the question of how agile are agile methods. To do this I synthesize seven general features of agility, drawing on management and sociology disciplines, into a framework, to act as a 'gold standard' by which to compare agile methods. I found that agile methods did not entirely measure up to this framework and that they were lacking in terms of (i) survival, (ii) prospering or thriving on change, and (iii) being able to regulate and leverage emotions in action responses to change. This paper offers: (i) a framework for assessing agility in software development, (ii) the elucidation of a knowledge gap in agile methods with respect to emotion, and, (iii) a conceptualization that reveals the need to incorporate emotional regulation and leverage into assessments of agility.

Keywords: agile methods, assessment, agency, emotion, game design

Introduction

This paper is a direct response to a current concern in the IS design literature: how agile are agile methods? In order to provide an answer, I propose I need some sort of 'gold standard' for agility, i.e. an independent benchmark to see how agile methods measure-up. When quality in software design is tested I have the likes of the Capability Maturity Model (SW-CMM) (Humphrey et al. 1991b), ISO 9001 (Paulk 1995), Bootstrap (Kuvaja et al. 1994), ami (Debou et al. 1994), SPICE (Dorling 1993), and TickIT (Classe 1993). However, when it comes to agility in software development, there are no independent standards or frameworks for assessing agility. Although Baskerville et al (2003) formulated metaprinciples for agile software development, these were based on software practitioner views; so the metaprinciples were somewhat in-bred, i.e. developed from within the context of agile methods. My study is different in that I propose a framework that is based on a synthesis of definitions of agility that borrow from outside the Information Systems Development (ISD) field, drawing on scholars in management and philosophy. I then use this framework to assess: (i) agile methods in an industrially context-free manner, i.e. based on general thinking about agile methods, (ii) agility in an utilitarian software development context (drawing on ISD literature), and (iii) in a computer games development context (drawing on a field study I conducted of a computer games studio in Singapore). As each assessment of agility is made, I continually reflect on what are useful ways of assessing agility in software development.

I suggest the need to probe deeply into a software development setting/team in order to perform such an assessment, since I argue that agility is largely based on the concept of human agency and emotional regulation. So, similarly to some quality assessment approaches that investigate organisational culture (e.g. (Humphrey et al. 1991b) to get a sense of a quality culture, we need in-depth field studies to get a sense of how agile the culture is. So, assessing agility does not just involve checking whether a team releases software in short cycles, or whether they reduce the cost of moving information between team members decision (Cockburn et al. 2001), but about understanding the deeper emotional profile of the developers in the setting. This paper's contributions are: (i) a framework for assessing agility in software development, (ii) the elucidation of a knowledge gap in agile methods with respect to psychological concepts such as emotion, and, (iii) a conceptualization that reveals the need to incorporate emotional regulation and leverage into agility assessments. The paper argues that agile methods can be improved if they were to embrace not just change per se (Beck 1999) but the emotional aspect of change.

Theoretical Foundations

Business Agility

The term agility in a business context was first coined by the Iaccoca Institute in 1991 (Sanchez et al. 2001). Generally, agility is often described as having the ability to adapt quickly to changing circumstances (Zain et al. 2003). Organisational agility is, "a

response to the challenges posed by a business environment dominated by change and uncertainty" (Zain et al. 2005:p83). Such a response may be characterised by the capability to survive and prosper in an environment of continuous and unpredictable change (Gunasekaran 1998). A capability, which in turn, according to Gunasekaran (1998) and Yusuf et al. (1999), stems from a flexible and adaptive capability. According to Shaw et al. (2005), agility is sometimes related to levels of organizational virtuality with strategically selected partners and supported by electronic commerce (Gunasekaran et al. 2002). So IT support for business processes and Inter Organisational Systems are also important features of being agile in a business context (Zain et al. 2005). However, Yusuf et al. (1999) explain that "the main driving force behind agility is change" (Yusuf et al. 1999):p34), i.e. as opposed to just the use of IT.

A Philosophy of Agility

Indeed, the Chambers dictionary would not disagree with the above interpretations of agility, although it does offer a new connotation when looking up the etymology of "agile", which it defines as 'nimble', i.e. moving quickly and lightly. The word agile is derived from the Latin *agere* – to do or act. So in one sense agile simply means to do something irrespective of the speed at which the doing occurs. However, if I take it to mean to act this has a more significant connotation, i.e. that a person takes action if they are agile. They do not stand still in the face of continuous and unpredictable change, but do something about it; they are flexible and adaptive (Yusuf et al. 1999). The opposite then would be someone who does not adapt or change their plans when faced turbulent times, i.e. escalating commitment to a failing course of action (Keil et al. 1993).
A person with the capability to act or adapt is referred to as an agent. In social theory taking action or acting is explored through the concept of action theory or agency. Anthony Giddens is a prominent author on the topic, who's view, in a nutshell, is that agents can make a difference in the world (Giddens 1984b). For Giddens, agency is "the stream of actual or contemplated causal interventions of corporeal beings in the ongoing process of events-in-the-world." (Giddens 1993:p81). The word 'interventions' is important since it indicates that agents have the capability to bring about a change in a situation. Just because a situation has been 'so' does not mean they should continue as such, "the seed of change is there in every act which contributes towards the reproduction of any 'ordered' form of social life" (Giddens 1993:p108). Thus an agent not only responds to change but is capable of provoking change. After all, "the main driving force behind agility is change" (Yusuf et al. 1999:p34)
Agents make their decision, whether wittingly or unwittingly, about how to act through their ongoing reflexive monitoring of themselves and their social contexts (Giddens 1984). Reflexive monitoring affords the agent a sense of ontological security; knowing what's what in any particular situation at any point in time. For example, what are social norms? What is socially acceptable? What is unacceptable? Do I care what is (un)acceptable? Agents rationalize their decisions and actions by forming a theoretical understanding of why they (are about to) act in certain ways. This in turn is guided by their so-called "projects" or overall plans or programmes in life, i.e. their motivations. Motivations in turn are directly connected to their emotions, "The connection of motivation to the affective elements of personality is a direct one, and is recognized in

everyday usage; motives often have 'names' – fear, jealousy, vanity, etc. – and these are at the same time commonly regarded as the 'names' of emotions." (Giddens 1993:p92). So feelings may also to some extent guide the actions of agents, such as not feeling like making a difference, being or stubborn, or indeed being gung ho and stepping up to adapt, survive and prosper when faced with change, or indeed, wishing to provoke change (Gunasekaran 1998). An extended version of Giddens' stratification model of the agent (Stacey et al. 2006) shows the mechanism underlying human agency. I will elaborate on this mechanism when I present my conceptualization.

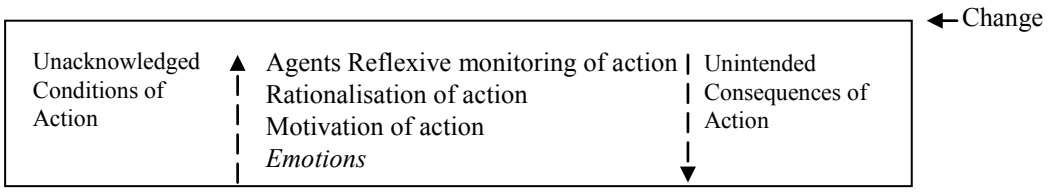


Figure 1: an extended version of Giddens' stratification model of the agent

General Features of Agility

Based on the above theoretical insights drawn from outside of agile software methods, and are therefore somewhat ISD independent, I now look across them all to try to tease out the essence of agility. I begin by summarizing and tabulating them into actions versus contexts (see table 1).

Table 1: Summary of views on agility

Agile actions	**Contextual forces enabling and constraining agility**	**Author**
ability to adapt quickly to	changing circumstances	Zain et al (2003)
a response to	challenges posed by a business environment dominated by change and uncertainty	Zain et al (2005)
the capability to survive and prosper in	an environment of continuous and unpredictable change	Gunasekaran (1998)
a flexible and adaptive capability		Gunasekaran (1998) and Yusuf et al. (1999)
	levels of organizational virtuality with strategically selected partners and supported by electronic commerce	Gunasekaran and Yusuf (2002)
	IT support	Zain et al (2005)
	Change	Yusuf et al. (1999)
To act or take action		Chambers dictionary
Exercise agency		Giddens (1993)
"the stream of actual or contemplated causal interventions of corporeal beings in the ongoing process of events-in-the-world"		Giddens (1993)
Regulate and leverage emotions for motivation and action		Stacey and Nandhakumar (2006)

Now that I have summarized these insights, I tease out the general features of agility, particularly based on the "agile actions" column – see table 2.

Table 2: General features of agility

General feature of agility		Contextual forces enabling and constraining agility
GF1	Be responsive	…a business environment marked by continuous, unpredictable, and uncertain change
GF2	Adapt quickly	
GF3	Survive	
GF4	Prosper	
GF5	Be Flexible	
GF6	Capability to intervene and exercise agency	
GF7	Capability to regulate and leverage emotions for action	
in…		

Expressing the general features of agility in table 2 as a statement:

> *Respond in a manner that exhibits quick adaptability, survival, prosperity, flexibility, intervention, and an ability to regulate and leverage emotions for action (being aspects of human agency), in a business environment marked by continuous, unpredictable, and uncertain change.*

Aspects of Agile Methods

Having proposed a set of 7 general features of agility (table 2), i.e. which were formulated independently of agile methods, I now review current thinking on agile methods.

Similarly to the above interpretations of agility in the management and philosophy literature, Agile software methods also emphasize flexibility (Cusumano et al. 1999), specifically in terms of, inter alia, the choice, sequence, scheduling, and documentation of development activities. Incarnations of agile software methods include eXtreme Programming (Beck 1999), Scrum (Schwaber 1995), Crystal (Cockburn 2002), Feature Driven Development (Palmer et al. 2002), the Rational Unified Process (Kruchten 1996), Dynamic Systems Development Method and Adaptive Software Development (Highsmith 2000). In these approaches, emphasis is placed on being responsive to disruptions in the environment such as changing user requirements, "Agility, for a software development organisation, is the ability to adopt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment" (Kruchten 2001:p27). Responsiveness is said to be enabled to some degree by developing in small iterations or sub cycles (Baskerville et al. 2003; Cockburn 2002); for example XP's 'iterations to release' phase (Beck 1999). Furthermore, XP has six phases; Exploration, Planning, Iterations to Release, Productionizing, Maintenance and Death (Beck 1999). Stories consisting of features are written and prioritized by the customer working alongside the developers (Exploration

and Planning phases) taking into account the effort estimates of the engineers, who then iteratively develop each feature (Iterations to Release phase). Many of XP's activities occur in close succession, doing a little of each activity at each iteration, i.e. a little planning, analysis, design and testing (Beck 2001). Continuous testing is an important aspect of XP's paired programming approach; as one writes code the other writes the test script (Beck 1999).

However, it is not just the implantation of customers in the development context (Baskerville et al. 2003) and an iterative approach that facilitates agility. It is also the way in which a team is organised, i.e. a software company needs to reduce the cost of moving information between people and reduce the elapsed time between making a decision to seeing the consequences of that decision (Cockburn et al. 2001), for example; by continually testing the software (Beck 1999). Reducing the cost of moving information between people may be achieved by placing developers physically closer, replacing documents with conversations and generally improving the team's sense of community and morale so that people are more inclined to relay valuable information quickly (Cockburn and Highsmith 2001:p131). According to Baskerville et al (2003), "Agile principles prioritize speed, responsiveness, and improvisation rather than quality or cost as traditional principles do. Contrary to traditional software development's emphasis on control, discipline, formality, and rigor, agile principles stress informal knowledge exchange, collaboration, and experience, and acknowledge more sensitivity to tailoring project practices to environmental conditions." (Baskerville et al. 2003):p75).

Aspects of agile software development

Table 3 summarizes the aspects of agile methods. Each aspect is assigned a code for easy reference later on. In the following section I compare these aspects back to those general features of agility already established in table 2.

Agile methods and general features compared

This section assesses the agility of those aspects of Agile methods reviewed above in light of the general features of agility; table four puts them side-by-side. There are three possible qualitative scores to be assigned to each aspect of agile methods; (i) maps well (to an general feature or GF), (ii) maps partially, (iii) no equivalent.

Do not map

Table 4 shows that GFs of survival (GF3) and prosperity (GF4) in a turbulent business environment are not addressed in agile software thinking. It is difficult to offer a rigorous reason for this, although the following section on applications of agile methods may offer one possible explanation, i.e. many studies of agile software development are conducted at large stable companies such as Microsoft or ABB. Survival is perhaps an issue more relevant to SME developers, such as the one featured in my empirical study. The general feature of prosperity means that developers thrive on uncertainty. Although Beck's article entitled "Embracing change with extreme programming" (Beck 1999)

would seem to reflect this, upon reading the article there is little to indicate the need to thrive on change.

Partially map

There are a few aspects that only partially map to the GFs. Having the capability to intervene and exercise agency (GF6) is only partially related to passing on information (A4) because relaying information may or may not evoke an agent's capability to make a difference. It will depend on the nature and impact of the information. Being able to regulate and leverage emotions for action (GF7) also only maps partially to improving the team's sense of community and morale (A4), because it does not address regulating (negative) emotions but focuses on positive emotions. According to Bagozzi (2003), the positive organisational behavior literature also shows how negative moods like embarrassment may have positive effects by provoking efforts to repair relationships, for example (Bagozzi 2003):p192).

Map well

However, many of the aspects of agile methods do map well with the GFs: (i) Being responsive to disruptions in the (A2:GF1), (ii) reducing the elapsed time between making a decision to seeing the consequences of that decision (A5:GF2), (iii) tailoring of project practices (A9:GF2), (iv) flexibility (A1:GF5), and (v) improvisation (A7:GF6).

Table 3: Aspects of Agile methods

Code	Feature	Author(s)
A1	Flexibility, with respect to the choice, sequence, scheduling, and documentation of development activities	(Cusumano and Yoffie, 1999)
A2	Being responsive to disruptions in the environment	Kruchten (2001)
A3	Develop in small iterations/ sub cycles	Cockburn (2002); Baskerville et al. (2003)
A4	Reduce the cost of moving information between people by - • Placing developers physically closer • Replacing documents with conversations • Improving the team's sense of community and morale so that people are more inclined to relay valuable information quickly	Cockburn and Highsmith (2001)
A5	Reduce the elapsed time between making a decision to seeing the consequences of that decision by – • continually testing the product • implanting customers in the development context to get quick feedback	Cockburn and Highsmith (2001), Beck (1999)
A6	Continuous testing	Beck (1999)
A7	Improvisation	Baskerville et al (2003)
A8	Experience	Baskerville et al (2003)
A9	Tailoring of project practices	Baskerville et al (2003)

The score

To heuristically assess how agile are agile methods, on the basis of table 4 one point is given for those aspects that mapped "well", half a point for those that mapped "partially" and no score is given for those marked as having either "no equivalent" or "doesn't map". The points are summed for each row and divided into the total number of aspects or rows in the table (denoted by # in left most column). There are 7 ASD aspects that map "well" with the GFs, 2 partially and 2 not at all, which renders a score of 8/11 or 72%. This is purely a heuristic, a gauge. The main message here is that agile methods do not entirely measure-up, in a qualitative sense, to the general features of agility.

Tailoring the General features for Agile software development

Table 4 demonstrated how aspects of agile methods (such as develop in small cycles) map to the general features of agility. I now refine and tailor the GFs to agile software development with those ASD aspects in table 4 that mapped well. It is important to do this to help make clear sense of the 2 subsequent assessments in this paper.

Table 4: Mapping aspects of Agile methods to general features of agility

#	GF Code	General features	ASD aspects	Score
1	GF1	Be responsive	Being responsive to disruptions in the environment (A2) *(Maps well)*	1
2			Develop in small iterations/ sub cycles (A3) *(Maps well)*	1
3	GF2	Adapt quickly	Reduce the elapsed time between making a decision to seeing the consequences of that decision (A5) *(Maps well)*	1
4			Tailoring of project practices daily (A9) *(Maps well)*	1
5	GF3	Survive	*No equivalent*	0
6	GF4	Prosper	*No equivalent*	0
7	GF5	Be Flexible	Flexibility, with respect to the choice, sequence, scheduling, and documentation of development activities (A1) *(Maps well)*	1
8	GF6	Capability to intervene and exercise agency	people are more inclined to relay valuable information quickly (A4) *(Maps partially)*	½
9			Improvisation (A7) *(Maps well)*	1
10			Experience (A8) *(Maps well)*	1
11	GF7	Capability to regulate and leverage emotions for action	Improving the team's sense of community and morale (A4) *(Maps partially)*	½
				= 8

Table 5: General features of agility for ASD

GF Code	General features for ASD
GF1	Be responsive: to disruptions in the environment, and develop in small iterations
GF2	Adapt quickly: reduce time between decision and consequence, and tailor practices daily
GF3	Survive
GF4	Prosper or thrive on change
GF5	Be Flexible: with respect to scheduling and documentation
GF6	Capability to intervene and exercise agency: be prepared to improvise and be experienced
GF7	Capability to regulate and leverage emotions for action

Assessing agile methods in an utilitarian context

The previous section assessed how agile are agile methods in a context-free manner. A concern at this stage is how valid is the agility framework (based on the set of 7 features) I devised? The following assessment is based on literature pertaining to utilitarian software development contexts, i.e. operating systems, productivity tools, web browsers, industrial automation and telecoms.

Cusumano et al. (1998) looked at Microsoft's "synchronize-and-stabilize" approach to software development, which "allows engineers to make a lot of changes in their designs until late in a project, while still keeping individuals synchronized and the evolving product more or less stable" (Cusumano and Selby 1998:ix). With respect to impact of context on practices, "Microsoft uses a range of build-cycle frequencies depending on the particular needs of a project and the amount of time required to complete a build successfully. Systems products generally take longer to build because of their size and the number of files and interdependencies included. Microsoft builds Excel, Word, and a test bed version of Office daily; it builds the full version of Office at least weekly" (Cusumano and Selby 1998:p275).

Baskerville et al's (2003) study of internet software development emphasized the impact of the rush-to-market working environment on the ASD process. In this context, there were frequent releases of the web browser, smaller feature sets, design and development occurred simultaneously, a dependence on development tools, an immediacy of feedback from the customer, product maintenance issues were ignored, and the methodology was tailored daily by the team (Baskerville et al 2003). In Cusumano and Yoffie's (1999) study of internet software development at Netscape and Microsoft, developers and their teams had a great deal of autonomy and a say in most aspects of the product development, i.e. the feature set. Furthermore, they were integral to the knowledge-ware of the product, self sustaining in terms of doing their own

process improvement, built their own tools, and required little motivation (Cusumano and Yoffie 1999).

Karlström et al. (2005) found that industrial automation and telecoms companies, which had implemented Agile, similarly benefited from the team reorganization aspects of Agile. The teams experienced improved communication, conversations proved to be more effective than documentation in resolving issues, and they obtained faster feedback by identifying and working closely with a customer representative. They also developed the most important feature first, which kept "the project deadline from affecting their scope. Only less important features might be scaled back or dropped" (Karlstrom and Runeson 2005:p46). Having ascertain and developed the most important feature first, their microplanning for remaining functionality involved ascertaining effort and trade-offs, i.e. if the customer asked for a new feature then the developer would ask them what ex tant feature they would like to drop. Table 6 structures and summarizes these agile software practices in an utilitarian software development context.

Table 6: Utilitarian agile software development practices

Code		Author(s)
	Build Frequency	
U1	Build frequencies at Microsoft vary with complexity and product size	Cusumano and Selby (1998)
U2	Frequent releases of web browser	Baskerville et al (2003)
	Design and Development	
U3	Make changes to designs until late in a project	Cusumano and Selby (1998)
U4	Design and development occur simultaneously	Baskerville et al (2003)
U5	Autonomy and influence over most aspects of the product design and development	Cusumano and Yoffie (1999)
	Features and Planning	
U6	Smaller feature sets	Baskerville et al (2003)
U7	Developed the most important feature first	Karlstrom and Runeson (2005)
U8	Microplanning	
	Tools	
U9	Reliance on development tools	Baskerville et al (2003)
U10	Tool building	Cusumano and Yoffie (1999)
	Customer Feedback	
U11	Immediate feedback from the customer	Baskerville et al (2003)
U12	Obtain faster feedback by identifying and working closely with a customer representative	Karlstrom and Runeson (2005)
	Process Structuring	
U13	Methodology was tailored daily by the team	Baskerville et al (2003)
U14	Team performs their own process improvement	Cusumano and Yoffie (1999)
U15	Product maintenance issues ignored	Baskerville et al (2003)
	Teamwork	
U16	Team requires little motivation	Cusumano and Yoffie (1999)
U17	Improved communication within the team	Karlstrom and Runeson (2005)
U18	Conversations more effective than documentation in resolving issues	Karlstrom and Runeson (2005)

Utilitarian ASD Practices and General Features Compared

I now assess the utilitarian ASD practices in light of the general features, and, as before, there are three possible qualitative scores to be assigned to each utilitarian ASD practice; (i) maps well, (ii) maps partially, (iii) no equivalent.

Do not map

Table 7 shows that GFs of survival (GF3) and prospering (GF4) in a turbulent business environment are not addressed in utilitarian agile software development. Although I proposed one explanation for this above, i.e. that many studies of agile software development are conducted at large stable companies such as Microsoft or ABB, there is always the possibility that these general features are not useful. Again, the GF of prospering or thriving on uncertainty is not evident in the practices.

Partially map

Tool building (U9 and U10) only map partially well to GF2 "adapt quickly" because it depends on what tools exactly have been built. They should not be the kind that lead to design lock-in but simply free you from tedious programming, i.e. code completion and custom libraries. Also, improved communication within the team (U17) partially maps to having the capability to intervene and exercise agency (GF6) because communication does not necessarily connote taking action; it may only be passive. Finally, a team that requires little motivation (U16) only partially connotes an ability to regulate and leverage emotions for action (GF7), since the mechanism by which motivation occurs is not explored.

Map well

However, many of the ASD practices do map well with the GFs, i.e. the build frequencies (GF1), the close connection between design and development (GF2), the quick feedback from customers (GF2), the flexibility to make late design modifications (GF5), prioritizing by developing the most important feature first (GF5), the daily tailoring of process (GF5), the autonomy of the developers (GF6), the microplanning (GF6) and that conversations trump documentation (GF6).

Table 7: Comparison of Utilitarian ASD practices and general features

#	GF	General features for ASD	Utilitarian ASD practices	Score
1	GF1	Be responsive: to disruptions in the environment, and develop in small iterations	Build frequencies at Microsoft vary with complexity and product size (U1) *(maps well)*	1
2	GF2	Adapt quickly: reduce time between decision and consequence, and tailor practices daily	Frequent releases of web browser (U2) *(maps well)*	1
3			Design and development occur simultaneously (U4) *(maps well)*	1
4			Reliance on development tools (U9) *(maps partially)*	½
5			Tool building (U10) *(maps partially)*	½
6			Immediate feedback from the customer (U11) *(maps well)*	1
7			Obtain faster feedback by identifying and working closely with a customer representative (U12) *(maps well)*	1
8	GF3	Survive	*No equivalent*	0
9	GF4	Prosper or thrive on change	*No equivalent*	0
10	GF5	Be Flexible: with respect to scheduling and documentation	Make changes to designs until late in a project (U3) *(maps well)*	1
11			Developed the most important feature first (U7) *(maps well)*	1
12			Methodology was tailored daily by the team (U13) *(maps well)*	1
13			Product maintenance issues ignored (U15) *(maps well)*	1
14	GF6	Capability to intervene and exercise agency: be prepared to improvise and be experienced	Autonomy and influence over most aspects of the product design and development (U5) *(maps well)*	1
15			Microplanning (U8) *(maps well)*	1
16			Team performs their own process improvement (U14) *(maps well)*	1
17			Improved communication within the team (U17) *(maps partially)*	½
18			Conversations more effective than documentation in resolving issues (U18) *(maps well)*	1
19	GF7	Be able to regulate and leverage emotions for action	Team requires little motivation (U16) *(maps partially)*	½
				= 15

The score

There are 13 utilitarian ASD practices that map "well" with the GFs, 4 partially and 2 not at all, which renders a score of 15/19 or 78%. This is purely a heuristic. The main message here is that ASD practices in an utilitarian software development context do not entirely measure-up to the general features of agility.

Refining the General features for Agile Software Development

Based on the above assessment, I may further refine the general features of agility for ASD. Again, those practices in table 7 that mapped "well", and are different to those already in the feature set, are incorporated. Table 8 constitutes a theoretical GF framework for assessing agility in software development.

Table 8: Refined General features of agility for ASD (a theoretical framework)

GF	General features for ASD
GF1	Be responsive: - to disruptions in the environment - develop in small iterations - build frequencies may vary with complexity and product size
GF2	Adapt quickly: - reduce time between decision and consequence - tailor practices daily - frequent releases - design and develop simultaneously - Work closely with the customer and obtain immediate feedback
GF3	Survive
GF4	Prosper or thrive on change
GF5	Be Flexible: - with respect to scheduling and documentation - accommodate design changes late into a project - develop the most important feature first - tailor practices daily - ignore product maintenance
GF6	Capability to intervene and exercise agency: - be prepared to improvise - be experienced - autonomy and influence over most aspects of product design and development, and process improvement - microplanning - conversations not documentation
GF7	Capability to regulate and leverage emotions for action

Knowledge Gap: Emotion

What the above theoretical assessments (tables 4 and 7) reveal is that the capability to survive (GF3), prosper or thrive on change (GF4), and foster emotional regulation and leverage (GF7) are all largely missing. Of these weakly represented general features I choose to explore GF7, since the theoretical foundations for this concept have already been laid. With respect to the emotion, although there is reference to agile methods improving a sense of community and team morale (Cockburn and Highsmith 2001) this does not go far enough, i.e. it does not explore how emotions make a difference to agility.

This lack of concern with the concept of emotion and inner life is supported by the philosophical foundations of Agile. According to (Brown et al. 2004) the Agile

approach is influenced by complex adaptive systems and soft systems thinking; the characteristics of living systems are applied to software (Highsmith 2002). This systems perspective is concerned with parts, their interaction, and how they connect to form a whole (Checkland 1999). Metaphorically, this regards the organization as an organism (Morgan 1997; Walsham 1993). The interacting parts of the 'organism' seek unity and homeostasis (a biological extension of the metaphor); trying to bring things back into rhythm in the aftermath of environmental change, connoting a joining, a synchronity, a rhythmical coordination of parts – circadian and routine. Such a biological perspective assumes software to be functionalistic; that it has parts and a whole have identifiable purposes and will adapt objectively. The subject, inner life and emotion are absent.

Addressing the Gap

To address the knowledge gap in ASD with respect to emotions, I now turn to an empirical assessment of how ASD practices fair in an emotive or "passionate" games development organisation. This is a novel context for studies of (agile) software development, and one which has received scant attention in the literature concerning ISD in general. The game design context is unique because games are designed to provide a compelling and sensory entertainment experience (Baba et al. 2001; Swartout et al. 2003), unlike a software application, which is usually designed to provide a business solution, enhance productivity or be "useful". A game is an intrinsically motivated phenomenon that produces experiences to be enjoyed for their own sake (Deci 1975). These experiences may also be described then as "inherently pleasurable", "exciting", and "enjoyable" (Calder et al. 1975).
The empirical assessment will draw and build on the general features for ASD as per table 8.

Research Setting and Method

I conducted an in-depth study of a computer games studio in Singapore, a country which has little in the way of natural resources and therefore relies on external trade and foreign direct investment to buttress its economy (Santiago 2003). In order to attract foreign direct investment, government agencies such as the Infocomm Development Authority (IDA) produce statistical reports and success stories which are presented at international games events such as E3. Local companies (games studios and otherwise) are encouraged and at the same time under some pressure to be successful so as to bolster the country's portfolio of achievements and hence its attractiveness. Time is of the essence in this agile economy, especially in the wake of economic and social crises such as the Asian Financial Crisis (1997) and SARS (2003). In 1998, for example, the Singapore government responded to the Asian Financial Crisis by setting up the Committee on Singapore's Competitiveness, which focused on high-tech entrepreneurship and innovation as a means of steering through the economic challenges (Wong 2002). Indeed, the computer games industry has been identified as one of the new 'engines' of Singapore's economy (Lim 2004).

The research approach adopted in this study is interpretive (Walsham 1993) involving a collection of detailed, qualitative data on the games development process in its context. I conducted twenty four semi-structured interviews with the team members of CGS (see table 9) over a three year period to understand their games development process and in what way it was agile. A typical project team at CGS involves six people: two programmers, two artists, a game designer and a project manager. CGS has developed numerous mobile games in association with studios in France and Italy, and their games have been distributed on European networks such as Vodaphone's.

Table 9. Interviews at CGS between January 2004 and March 2006

Interview with	Number of interviews
Team	1
Project Manager	2
Game Designer	5
Programmers	12
Artists	4
TOTAL =	24 (6 team members x 4 visits)

Data collection

From January 2004 to March 2006 I made several visits to CGS to conduct interviews with project team members. Detailed notes were kept on the interviews and the observations of work practices during each visit. Company documents and flowcharts were also analyzed to gain more insight into their software practices and to verify interview notes. I have reason to believe that their practices were influenced by agile methods since they expressed an awareness of them and some had used them in previous occupations. By adopting an interpretive approach to collecting and analyzing the interview data, I attempted to understand the phenomenon (the games development process) through the meanings that team members assigned to that process. Such meanings may be rooted in data incidents like jargon, symbols, and metaphors the interviewee uses. It is up to the researcher to explore these 'incidents', taken-for-granted understandings (Schutz 1967) within the setting.

The data collection process was as follows: (i) I began by interviewing the team as a whole so as to become better acquainted, introduce my research project to them and to see how the related to each other in that situation, (ii) semi-structured interviews were then conducted with all six team members on four visits to CGS during the past three years, in which I asked a combination of direct and open-ended questions, and (iii) a summary of the data collected was periodically presented to the Managing Director for him to validate and provide us with feedback.

The direct questions were asked with reference to specific aspects of their practices (see table 10). These questions were based on those used by (Radice et al. 1999) in their programming process study. Whilst their study was not used to assess agile practices,

the questions they asked in their interviews with programmers are still evocative of many aspects of Agile. To show how these questions related to the GF framework in table 8, I provide the column "pertains to GF" in table 10. General features of agility GF3, GF4 and GF7 are not covered by these questions. However, they are addressed in subsequent sections that deal with the more unstructured data I collected on their practices.

Table 10: Structured questions pertaining to aspects of ASD

Structured Question	**Pertains to GF**
SQ1. What is your Build frequency?	GF1
SQ2. Do you find yourself following any procedures?	GF6
SQ3. How do you estimate resources, i.e. manpower…?	GF5
SQ4. How are changes controlled?	GF5
SQ5. What development tools do you use and build?	GF2
SQ6. What checks and balances, reviews, feedback and approvals are involved?	GF2
SQ7. What is the level of customer awareness in your unit/team?	GF2
SQ8. Are there any defined procedures for your work?	GF6
SQ9. What documentation do you use?	GF6
SQ10. What documentation do you produce?	GF6

Data Analysis

To organize the mass of data and prepare to analyze it, the interview transcripts and field notes were imported into a software package 'nVivo'. To structure my analysis of the unstructured data I collected, I drew upon Miles and Huberman's three tier coding approach, i.e. descriptive, interpretive and pattern coding (Miles et al. 1994). Firstly, I performed descriptive coding in which I focused on identifying agile software practices and related activities; for example, the practice of continuous play-testing. Secondly, I performed interpretive coding to put these practices and related activities into context; I wanted to understand the local dynamics pertaining to these activities, i.e. why and how they occurred. Thirdly, I performed pattern coding, in which I looked across all the practices in context to discern a pattern in their occurrence, i.e. emergent themes. As patterns and themes emerged, I began to think of them in more abstract terms and develop my ideas on ASD in a games development context.

Case Description and Analysis

CGS has a management hierarchy with four layers of managers: senior management, middle-management, project leaders and executives. The working environment at the premises gave a sense of home-from-home: there was a meeting room with antique Indonesian furniture which doubled as the Game Designer's office, an office for the programmers who seemed subdued although their walls seemed to reflect their female fantasies, an office for the artists who had various traditional games like table football

set up, and a laboratory full of computers for rendering purposes. The environment seemed conducive to long hours of work. My study focused on the work practices of the Game Designer, Project Manager, Lead Programmer, Programmers, and Artists. These were the dominant organizational actors involved in all games projects. However, sometimes these roles were not discrete and overlaps occurred. The Lead Programmer sometimes became involved in game design and the Game Designer was also the Managing Director, which meant he had to participate in business 'events' outside the organizational context; with government agencies, partners, clients and distributors.

Structured data relating to agility in CGS' practices

Table 10 provided a list of direct questions that I asked in each interview, which retrospectively are evocative of many aspects of Agile methods. In table 11 I present a summary of the responses the participants gave to them, assign a code to them, and

Table 11: Comparing game design practices to the GF framework

#	GF	General features for ASD	Game design practices	Score
1	GF1	Be responsive: - to disruptions in the environment - develop in small iterations - build frequencies may vary with complexity and product size	CGS have daily build for code units in early stages of project, and they have daily builds for entire game code towards end of project as units are integrated and play-testing commences *(maps well)*	1
2	GF2	Adapt quickly: - reduce time between decision and consequence - tailor practices daily - frequent releases - design and develop simultaneously - Work closely with the customer and obtain immediate feedback	CGS use the following tools: Jcreator, WTK2.0, J2ME, Java SDK, 3Dstudio, phone emulator *(maps well)*	1
3			I don't build tools *(partially maps)*	½
4			To obtain feedback on progress CGS review milestone completion, hold daily and weekly meetings, hold code reviews. *(maps well)*	1
5			Not much customer awareness at CGS: "I are gamers so I are also the customer", "depends on target audience for the game", "most are aware except the more junior members", "not much awareness since most of my games are produced for "suits" at Telcos" *(doesn't map well)*	0
6	GF3	Survive	*No equivalent*	0
7	GF4	Prosper or thrive on change	*No equivalent*	0
8	GF5	Be Flexible: - with respect to scheduling and documentation - accommodate design changes late into a project - develop the most	estimating resources is either based on experience, depends on the complexity of the game, or by asking individuals to estimate their own time *(maps well)*	1

		important feature first - tailor practices daily - ignore product maintenance		
9			Changes are controlled by: decision-makers, or not strictly tracked or controlled or are by the design committee *(doesn't map)*	0
#	**GF**	**General features for ASD**	**Game design practices**	**Score**
10	GF6	Capability to intervene and exercise agency: - be prepared to improvise - be experienced - autonomy and influence over most aspects of product design and development, and process improvement - microplanning - conversations not documentation	As opposed to procedures I just follow the project lead *(doesn't map)*	0
11			estimate resources by asking individuals to estimate their own time *(maps well)*	1
12			CGS has little in the way of defined procedures except for the concept stage. Sometimes its up to the project leader, otherwise developers are autonomous *(Maps well)*	1
13			CGS' only commonly used document is the Game Design Document (GDD) *(maps well)*	1
14			CGS produces the following documents: technical doc, bug log, game design doc, meeting notes, reports, marketing docs *(doesn't map)*	0
15	GF7	Be able to regulate and leverage emotions for action	*No equivalent*	0
				= 7.5

The score

There are 7 agile games development practices that map "well" with the GFs, 1 partially and 7 not at all, which renders a score of 7.5/15 or 50%.

What does and does not map

What maps particularly well here are the build frequencies CGS uses (GF1), the tools they use (although they do not build tools), the frequent status meetings and code reviews (GF2), their resource estimation approach (GF5), reliance less on procedure and more on improvisation (GF6), little reliance on document (GF6). What is at odds with the GFs for ASD is the over creation of documentation that is hardly used (GF6), lack of attention to emotional dimensions of team work (GF7), there are times when team members rely on the leader (GF6), there is no evidence of thriving on change (GF4), nor survival instincts, i.e. heroic efforts (Bach 1995). More controversially is the lack of customer awareness (GF2). GF2 states the importance of working closely with customers in order to obtain quick feedback so as to adapt quickly and reduce the time between decision and consequence. However, the developers insisted that they were also "gamers" and therefore were in-tune with the needs of external customers. This issue in itself requires further scrutiny to understand its validity.

Data sufficiency

ASD practices in the games development context seem to fair worse based on the heuristic score than the utilitarian ones. However, this is due to the way the data was collected. To assess agility I need to collect much richer data; it is insufficient to ask direct questions only. Given I conducted semi-structured interviews, there was ample opportunity for the interviewee to take the "reins" of the conversation and talk about their games projects and practices. The next section presents narrative data which reveal evidence for those GFs for which CGS scored zero above, i.e. GF3, GF4, GF6 and GF7.

Narrative: "against the odds"

The Game Designer (Alf) and team began work on the concept for a sci-fi game for which their investor gave them five months to develop. On the team was Jacky, who was also the chairman of the Animé Club and had his own artists who were already working on a futuristic animation series. Alf suggested merging their ideas but Jacky did not want to since he had his own animation team and concept already. Jacky shared these ideas with Alf and they decided that Jacky would work on an animation series whilst Alf would produce a PC game based on these ideas. After a month or so into the production phase they began discussing how to share the intellectual property. Alf was surprised when this led to a disagreement and Jacky renouncing their partnership and Jacky's staff "walking out". Although this was a serious blow to the project since the game ideas were Jacky's, Alf did not want to give up, particularly since he had already made a commitment to MDA. Alf persevered and had to decide whether to come up with a completely new game idea or continue with what he had so far but alter the

storyline. He decided to do the latter and the game Beyond Event Horizon began to take shape. On reflection, Alf said he was glad that there had been a parting of ways because it became clearer and easier to maneuver. As things started moving again, his senior 3D artist was poached by a US game company. After this second setback Alf decided to coordinate the art team himself, which he described as "a nightmare". However, he mobilised an "army" of freelance 2D and 3D artists. In the end the project was deemed a success; they were invited to Austin, Texas in the US to pitch to a publishing company.

Analysis

What this narrative reveals is a spirit of survival, a prospering or thriving on change, and the regulation and leveraging of emotions. For instance, Jacky and his team's departure from CGS led to the breakdown of established routines, and resulted in the incumbents losing their sense of security, or using Giddens' phrase, a loss of ontological security, i.e. what's what? However, Alf's rationalisation for getting the project back on track was two-fold; he had a sense of duty to since the Media Development Authority was funding the project, and also the project excited him. His motivation for steering back the project was therefore linked to his feelings of excitement, loyalty and obligation. These feelings provoked him to adjust his reflexive monitoring, or awareness of the situation, so as to identify and draw on new rules and resources that would support his improvisations, i.e. modifications to the game's story and concept, which were originally provided by Jacky. Faced with another unexpected environmental change - the resignation of the lead 3D artist and some of his colleagues - Alf took over the reins of lead artist. Through his continuous monitoring of contexts of human resources he assembled a new team of artists. He would not give up. His capability to act and to overcome was related not just to his ability to monitor and mobilise his contexts however, but to his emotional attachment to the project; his motivation was emotional.

Furthermore, it would seem that his survival instinct was related to quick adaptation, responsiveness and flexibility. This suggests that the GFs of agility being responsive, adapt quickly and be flexible may be all subsumed under this GF, i.e. under survival (GF3). So one possible reason why there had been no direct evidence for the general feature of survival is that it is a meta general feature.

Refining CGS's scores

If I 'factor' in evidence of GF3, GF4 and GF7 into table 11, i.e. assign a score of "1" to each of these general features, then CGS's score improves from 7.5 to 10.5/15, or 70%.

Discussion and Implications

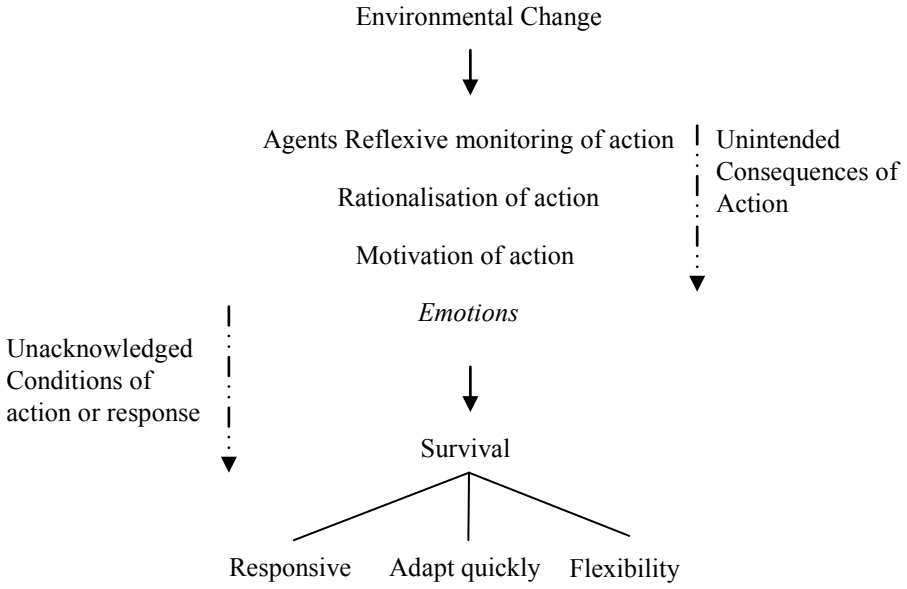
Based on the theory and empirically based assessments, I present my conceptualization of Affective Agility (figure 2); affective being evocative of emotions and moods. In order to arrive at this conceptualization I translated the general features into concepts, with the exception of GF4 – prospering or thriving on change. This seems too general to

be of use in the conceptualization. As per my analysis of the narrative, "survival" is now a meta concept for being responsive, adapt quickly and flexibility (the scores of the agility assessments would improve in this case). The emotional aspect of this conceptualization is based on my empirical study of a games development context, which demonstrated the appropriateness of incorporating this concept.

Conceptualization

Drawing on my data analysis and theoretical foundations, I depict my understanding or conceptualization of Affective Agility.

Figure 2: Conceptualization of Affective Agility



Starting from the top of figure 2, environmental change is a key driving force of agility (Yusuf et al. 1999). From a sociological perspective this can be interpreted as provoking ontological insecurity or concerns over what constitutes one's reality, as characterised by norms and routines for example. Developers become aware of such changes (e.g. new features requested by the customer) through their reflexive monitoring of their contexts. They then interpret and rationalize the meaning of the disruption (Kruchten 2001) or change. As they do so this affects their motivation for action and produces a mixture of emotions, which are unintended consequences of those changes. These emotions then become part of the unacknowledged conditions for a developer's action. In their practical consciousness, i.e. at a tacit level, the developer's or agent's motivation to respond to change leads them to reinterpret and re-rationalize the meaning of the changes in their context, which in turn produces a positive mood of and desire to survive. As my empirical study showed, survival is a meta concept, and this entails being responsive, adapting quickly and being flexible in the face of change. Emotions therefore mediate and provoke the action responses to change, as per being responsive,

adapting quickly and being flexible; or, in terms of agile practices: build frequencies (Cusumano and Selby 1998), the time taken between making a decision to seeing its consequences (Cockburn et al. 2001), the tailoring of project practices (Baskerville et al. 2003), making changes to designs until late in a project (Cusumano and Selby 1998), and developing the most important feature first (Karlstrom and Runeson 2005), for example.

I have made three contributions to knowledge in this paper: (i) a framework for assessing agility in software development; one was suggested in table 8, (ii) the elucidation of a knowledge gap in agile methods with respect to emotion, and (iii) a conceptualization that reveals the need to incorporate emotional regulation and leverage into agility assessments; I depicted this in figure 2.

Implications and Future Research

My conceptualization of Affective Agility has implications for how I may assess agile methods or agility in a software development setting. Since, as I have shown, agility is a chronically emotional concept, attempts to assess how agile are agile methods and agile practices could consider incorporating assessments of emotional regulation and leverage in the development context under investigation. This requires in depth longitudinal field studies such as ours to gain such evidence; in similar fashion to some quality assessment approaches that investigate organisational culture (e.g. Humphrey et al. 1991b). So, assessing agility does not just involve checking whether a team releases software in short cycles and so forth, but it is about understanding the deeper emotional profile of the developers in the setting. So, to elaborate on GF7 of agility, "Be able to regulate and leverage emotions for action", the researcher needs to encourage the interviewee to talk about incidents that were perhaps "trying" or "painful", and observe the content and manner in which they recount their story. Better still would be for the researcher to engage in participant observation and spend extended periods of time in the development setting so as witness such incidents first hand (Nandhakumar et al. 1997). Although I have found emotion to be a key aspect of agility within the context of games development, I do not generalize beyond this context but invite further studies to assess the significance of emotion in mediating agility.

My research also has implications for practitioners of agile software development. According to my findings, to be agile in a software development context is not solely about your build frequencies, whether you implant customers into your context, and placing conversation over documentation. Its also about having people with emotional intelligence, who know not to cover-up their mistakes, or bask in confusion, but are able to take heroic steps to bring issues out into the open and deal with the conflictual consequences of doing so, "the development of software systems is a complex, sociotechnical process in which demanding technical challenges are confronted within a conflict-laden political context" (Wastell 1996:p29). The source of such conflict may be people's different interests, goals, world views (Newman et al. 1990; Orlikowski 1993), as well as the social structures (Walsham 1993) identified with the workplace. So, an aspect of agility in software development is developers regulating and leveraging the

emotional consequences of conflict in their settings. Managers may therefore try to cultivate good emotional regulation, or indeed train developers not just on technical skills but emotional skills too.

Conclusions

This paper has focused on addressing the question how agile are agile methods. To do this I initially synthesized seven General Features (GFs) of agility, drawing on management and sociology disciplines, into a framework, to act as a 'gold standard' by which to compare agile methods. When I compared the general features to studies on Agile thinking, I found that agile methods did not entirely measure up to this framework; they were lacking in terms of survival (GF3), prospering or thriving on change (GF4) and being able to regulate and leverage emotions in action responses to change (GF7). I found the same thing when I used this framework to analyze studies on Agile methods within an utilitarian software context. Of the three weakly represented general features I chose to explore GF7, since the theoretical foundations for this concept had already been laid. This represented an opportunity to elaborate on the role of agency and emotion in agility. To this end I presented data from a field study of a computer games development organisation, because this is an emotive and sensory (Baba et al. 2001) software development setting. Although some of the agile practices were left wanting, such as over-production of documentation, I did find that emotion played an important role in priming developers for action. On the back of this finding I presented a conceptualization of Affective Agility (figure 2), i.e. that emotion plays an integral role in an agile setting. Therefore, GF7 (emotional regulation) was justified as an important part of the Agility Assessment Framework (AAF). At the same time throughout the assessments I conducted, I embellished the framework by elaborating on the general features with the agile practices and thinking that mapped well. Although the AAF (table 8) may be useful, it is also rather prescriptive. To assess how agile are agile methods in a particular context I suggest reference to and the adaptation of my Affective Agility model (figure 2). Such assessment exercises require in-depth longitudinal field studies to gain an understanding of the emotions and their implications for agility in the context.

References

Baba, Y., and Tschang, F.T. "Product development in Japanese TV game software: The case of an innovative game," International Journal of Innovation Management (5:4) 2001, pp 487-515.

Bach, J. "Enough About Process: What I Need are Heroes," IEEE Software (12:2) 1995.

Bagozzi, R.P. "Positive and Negative Emotions in Organizations," in: Positive Organizational Scholarship: foundations of a new discipline, K.S. Cameron, J.E. Dutton and R.E. Quinn (eds.), Berrett-Koehler Publishers, San Francisco, 2003.

Baskerville, R., Ramesh, B., Levine, L., Pries-Heje, J., and Slaughter, S. "Is "Internet-speed" software development different?," Software, IEEE (20:6) 2003, pp 70-77.

Beck, K. "Embracing change with extreme programming.," IEEE Computer (32:10) 1999, pp 70-77.

Brown, R., Slinkman, C., and Nerur, S. "Philosophical Shifts in Software Development," Proceedings of the Tenth Americas Conference on Information Systems, August 2004, New York, 2004.
Calder, B.J., and Staw, B.M. "Self-perception of intrinsic and extrinsic motivation.," Journal of Personality and Social Psychology (31) 1975, p 599–605.
Checkland, P. Systems Thinking, Systems Practice John Wiley and Sons, LTD, Chichester, 1999.
Classe, A. "TickIT update.," Software Quality Management, (19) 1993, pp 10-11.
Cockburn, A. Agile Software Development Addison-Wesley, Boston, 2002.
Cockburn, A., and Highsmith, J. "Agile Software Development: The People Factor," IEEE Computer (2001:November) 2001.
Cusumano, M., and Selby, R.W. Microsoft's secrets: How the world's most powerful softwre company creates technology, shapes markets and manages people Simon and Schuster, NY, 1998.
Cusumano, M.A., and Yoffie, D.B. "Software development on Internet time," Computer (32:10) 1999, pp 60-69.
Debou, C., Kuntzmann-Combelles, A., and Rowe, A. "A quantitative approach to software process management," Software Metrics Symposium, 1994., Proceedings of the Second International, 1994, pp. 26-34.
Deci, E.L. Intrinsic motivation Plenum, New York, 1975.
Dorling, A. "SPICE - Software Process Improvement and Capability dEtermination," INFORMATION AND SOFTWARE TECHNOLOGY (35:6-7) 1993, pp 404-406.
Giddens, A. The Constitution of Society: Outline of the Theory of Structure. University of California Press, Berkeley, CA., 1984.
Giddens, A. The nation state and violence Polity Press, Oxford, 1984b.
Giddens, A. The Giddens Reader Macmillan, London, 1993.
Gunasekaran, A. "Agile manufacturing: enablers and an implementation framework.," Int. J. Prod. Res. (36:1223–1247) 1998.
Gunasekaran, A., and Yusuf, Y.Y. "Agile manufacturing: A taxonomy of strategic and technical imperatives.," Int. J. Prod. Res. (36) 2002, p 1357–1385.
Highsmith, J. Agile Software Development Ecosystems Addison-Wesley, Boston: MA., 2002.
Highsmith, J.A. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems Dorset House Publishing, New York, NY, 2000.
Humphrey, W.S., Snyder, T.R., and Willis, R.R. "Software Process Improvement at Hughes-Aircraft," Ieee Software (8:4), Jul 1991b, pp 11-23.
Karlström, D., and Runeson, P. "Combining Agile Methods with Stage-Gate Project Management," IEEE Software (May/June) 2005.
Keil, M., and Mixon, R. "Laboratory Studies of IS Failure as Escalating Commitment to a Failing Course of Action: Overcoming the Obstacles," 14th International Conference on Information Systems (ICIS), Orlando, FL, 1993, p. 382.
Kruchten, P. "A Rational Development Process," Crosstalk (9:7) 1996, pp 11-16.
Kruchten, P. "Agility with RUP," Cutter IT Journal (14:12) 2001, pp 27-33.
Kuvaja, P., and Bicego, A. "BOOTSTRAP - a European assessment methodology," Software Quality Journal (3:3) 1994, pp 117-128.
Lim, T. "The Gaming Landscape," Asia Games Conference, Singapore, 2004.

Miles, M.B., and Huberman, M.A. Qualitative Data Analysis Sage Publications, Thousand Oaks, 1994.
Morgan, G. Images of Organization Sage, 1997.
Nandhakumar, J., and Jones, M. "Too Close for Comfort? Distance and Engagement in Interpretive Information Systems Research.," Information Systems Journal (7) 1997, pp 109-131.
Newman, M., and Noble, F. "User Involvement as an Interaction Process: A Case Study," Information Systems Research (1:1) 1990, pp pp. 89-113.
Orlikowski, W.J. "CASE tools as organisation change: investigating incremental and radical changes in systems development," MIS Quarterly (17:3) 1993, pp 309-340.
Palmer, S.K., and Felsing, S.M. A Practical Guide to Feature Driven Development Prentice Hall, Upper Saddle River, NJ., 2002.
Paulk, M.C. "How ISO 9001 compares with the CMM," Software, IEEE (12:1) 1995, pp 74-83.
Radice, R.A., Harding, J.T., Munnis, P.E., and Phillips, R.W. "A programming process study.," in: IBM Systems Journal, IBM Corporation/IBM Journals, 1999, p. 297.
Sanchez, L.M., and Nagi, R. "A review of agile manufacturing systems.," Int. J. Prod. Res. (39) 2001, p 3561–3600.
Santiago, T. "Hungry Singapore looks to move up food chain," in: Electronic Engineering Times, 2003, p. 39.
Schutz, A. Phenomenology of the Social World. Northwestern University Press, Evanston, IL, 1967.
Shaw, N.E., Burgess, T.F., de Mattos, C., and Stec, L.Z. "Supply chain agility: the influence of industry culture on asset capabilities within capital intensive industries," International Journal of Production Research (43:16) 2005, p 3497–3516.
Stacey, P., and Nandhakumar, J. "Responding to games development challenges through mood-mediated improvisation," ECIS 2006, Goteborg, Sweden (forthcoming), 2006.
Swartout, W., and van Lent, M. "Making a game of system design," Communications of the ACM (46:7) 2003, pp 32--39.
Walsham, G. Interpreting IS in Organisations John Wiley, Chichester, 1993.
Wastell, D.G. "The fetish of technique: methodology as a social defence," Inform Syst J (6:1), January 01, 1996 1996, pp 25-40.
Wong, P.K. "From Using to Creating Technology: The Evolution of Singapore's National Innovation Systems and the Changing Role of Public Policy," in: Technology Policy in East Asia, S.e.a. Lall (ed.), World Bank and Elgar Press, 2002.
Yusuf, Y.Y., Sarhardi, M., and Gunasekaran, A. "Agile manufacturing: the drivers, concepts and attributes.," Int. J. Prod. Econ. (62) 1999, p 33–43.
Zain, M., Kassim, N.M., and Mokhtar, E. "Use of Information Technology and Information Systems for Organisational Agility in Malaysian Firms," Singapore Management Review (25:1) 2003, pp 69-83.
Zain, M., Raduan, C.R., Iskandar, A., and Maslin, M. "The relationship between information technology acceptance and organizational agility in Malaysia," Information & Management (42) 2005, p 829–839.