A Model of Distributed Agile Team: Agent-based Modeling approach

Submission Type: Research-in-Progress

Yoon Sang Lee
University of Texas at Arlington
yoonsang.lee@mavs.uta.edu

Abstract
Due to high failure rates of traditional plan-driven software development methodologies, Agile Software Development was suggested as an alternative. Many studies suggested that Agile has many characteristics of Complex Adaptive System (CAS). Until now, most of the findings from previous studies are about discrete elements of Agile methodology or episodic events, and they provide microscopic view or theoretical background of the Agile methodology. However, Agile methodology and its development teams need to be understood as a whole in the macroscopic view. In this study, we suggest an Agent-based Model of remote Agile team as the tool for simulation of the mechanism of distributed Agile team and the effect of communication on the distributed Agile team. Through this Agent-based Model, the research on Agile team will be able to equip a macroscopic view of Agile team using bottom-up approach under controlled team conditions.

Keywords
Agent, Agent based modeling, Agile, software development.

Introduction
During past several decades, information systems have become an integral part of business, and software development has become the core part of information system development as a business enabler. Various software developments methodologies have been suggested, the most widely accepted and used in software development projects is the plan-driven waterfall methodology. Augustine et al. (2005) characterized traditional software development methodologies as the ones reflecting linear, sequential processes, and this methodology is expected to provide a good management functionality (Nerur et al. 2005). The traditional software development methodology was expected to provide higher management ability for software development, but software development projects have showed high failure rates: 49% projects had budget overruns, 47% had higher-than-expected maintenance costs, and 41% failed to deliver the expected business value and ROI (Asay 2008). Most of these problems were due to the characteristics of business requirements which include frequent change of requirements, unknown solution, and unanticipated effect (Asay 2008; Augustine et al. 2005).

In 2001, the Manifesto of Agile Software Development was suggested as an alternative to traditional plan-driven software development methodologies to cure the high project failure rate. According to the Agile Manifesto, Agile methodology emphasizes human-oriented and adaptive-to-change characteristics. Meso and Jain(2006), Nerur et al. (2005), and Augustine et al. (2005) suggested that Agile has many similarities with Complex Adaptive System (CAS). The important common characteristics of CAS and Agile methodology are the interaction and relationship among members in a team and the iterative approach to the goal of the development project. Especially, the agile principle emphasizes the face-to-face interaction among software developers and between customers and developers.

Although Agile methodology appears as a viable alternative to traditional software development methodologies, the application of Agile to large scale projects is still questionable because of the
uncertainties of Agile projects caused by the characteristics of CAS. In addition, current trend of the remote software development environments such as offshore software development or distributed software development team adds another uncertainty to the use of Agile methodology because Agile methodology requires frequent interactions among team members and between customers and developers.

In previous studies, researchers have studied the theoretical background of Agile, CAS–related characteristics of Agile methodology, and the critical success elements for Agile projects. Most of the findings from these studies are discrete elements of Agile methodology or episodic events, and they provide microscopic view or understanding of the Agile methodology. However, Agile methodology and its development teams need to be understood as a whole in the macroscopic view because it is an instance of CAS. However, it is hard to equip a holistic and precise theoretical logic concerning the bottom-up linkage from individual to collective-level with traditional approaches (Nan 2011).

Moreover, finding enough number of teams that have various aspects in accord with the interest of a study or conducting the large number of experiments that are required to study the related variables in a study is not easy for team level research. Also, it is hard to eliminate the coinciding effects that a researcher doesn’t want to include in his/her study. As an example related to Agile, Hewstone et al. (2002) suggested that the face-to-face interactions increase the favoritsm among the team members in the same physical location. However, as cited in Nan et al. (2008) other studies such as Finn et al. (1997), Burke et al. (1999), Burke and Chidambaram (1999), Burgoon et al. (2002), and Panteli and Davison (2005) suggested that there is no difference on the team performance by the difference of the degrees of fact-to-face interaction.

As we suggested earlier, Agile methodology has the characteristics of CAS. Generally, the evaluation of the output from the complex system is considered difficult due to its complex interactions among the elements in the system and the iteration of similar but different processes. In this study, we suggest an Agent-based Model of remote Agile team as the tool for simulation of the mechanism of distributed Agile team and the effect of communication on the distributed Agile team because it can satisfy various requirements of this study such as bottom-up approach, macroscopic view of Agile team, control of coincident effect, and controlled team conditions.

This study attempts to contribute to the IS research of Agile by proposing a basic Agile team model and by answering two questions regarding the effect of the easiness of interaction in distributed Agile team setting by using Agent-based Modeling technique. The two questions are:

1. Is there any performance difference between remote team and collocated team?
2. Is there any performance difference between two different remote teams by the difference of the degree of media richness?

The structure of this article is as follows: First, we survey the theoretical background of Agile team as an instance of CAS to validate the application of Agent-based Modeling to Agile team mechanism. In this section we look at the tasks of Agile methodology and the mapping between the tasks in Agile methodology and CAS principles. Second, we find the characteristics of distributed Agile team to add them into the Agent-based Model. Third, we provide behavioral rules and interactions for Agent-based Model based on previous literature. In the last part, since this paper is research-in-progress, future experiments and analysis plan are discussed.

**Literature Survey**

**Literature on Agile Method**

Until 90’s the main stream of software development methodology was plan-driven waterfall methodologies. But the software development projects that used plan-driven methodology had a chronic problem of cost overruns, overdue, and slow responsiveness from the software change requirement (Lee and Xia 2010; Boehm 2002, Larman 2004). As a remedy for the problems of traditional software development methodology, Agile method has been suggested, and Extreme Programming (XP), Scrum, Feature-Driven Development (FDD), Dynamic Systems Development Method (DSDM) are commonly used in development projects (Lee and Xia 2010; Meso and Jain 2006).
In 2001, 12 principles of agile development were proposed with the publication of Agile manifesto. Agile method has emphasis on small team, face-to-face interaction, collaboration and communication, adaptive software, responsiveness, tacit knowledge, self-organized team (encourage role interchangeability), informal communication, organic form, and evolutionary-deliver model (Nerur et al 2005; Nerur and Balijepally 2007). Typical Agile project is carried out in a small size team with face-to-face interaction among team members through people driven process using pair programming (Bose 2008). Bose(2008) represented the steps in Agile method of software development in Figure 1. As seen in the Figure 1, Agile method goes through multiple iterations. In one iteration, customers' requirements are delivered to software developers, and the modification, test, and small release of software are conducted accordingly. This one set of tasks in one iteration. This set of task is repeated in multiple iterations until the software development is completed. Pair Programming is used as a programming method, and during the programming interaction between team members and managers happens frequently. As iterations continue, individual team member's development skill and domain knowledge are increased, and, as a result, the entire team comes to reach the solution for the complex information system task.

These typical characteristics of Agile methodology reflect the essential characteristics of complex adaptive system (Nerur et al. 2005). Meso and Jain(2006) suggested mapping table of Agile practice and CAS principles. Through the mapping of Agile to CAS we may have the understanding of the types of interdependencies between individuals, organizations, and IS solution in Agile software development (Meso and Jain 2006).

<table>
<thead>
<tr>
<th>Agile Practice</th>
<th>CAS Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequent releases and continuous integration</td>
<td>Principle of growth and evolution</td>
</tr>
<tr>
<td>Need for frequent feedback</td>
<td>Principle of transformative feedback loops</td>
</tr>
<tr>
<td>Proactive handling of changes to the project requirements</td>
<td>Principle of emergent order</td>
</tr>
<tr>
<td>Loosely controlled development environment</td>
<td>Principle of distributed control</td>
</tr>
<tr>
<td>Planning kept to a minimum</td>
<td>Principle of growth and evolution / Principle of emergent order</td>
</tr>
<tr>
<td>Enhancing continuous learning and continuous improvement</td>
<td>Principle of growth and evolution / Principle of interactions and relationships</td>
</tr>
<tr>
<td>Emphasis on working software product</td>
<td>Principle of path of least effort (based on Zipf, 1949)</td>
</tr>
</tbody>
</table>

**Table 1. Mapping Agile Practice and CAS Principles (Meso and Jain 2006)**

**Literature on Distributed Agile Development**

One of the emphases of Agile method is the face-to-face interaction. Especially, when customers and multiple developers are collocated and interact with each other, the benefits of Agile method are realized (Boehm and Turner 2005). However, offshore agile development in Moore and Barnett (2004) and remote Agile development in Sepulveda (2003) suggested that distributed Agile development has potential for the distributed development method. Even if the two studies suggested the positive possibility for distributed Agile methodology, there are still many factors to consider. Bose (2008) suggested 6 critical strategies by performing 12 case studies. They are convenient communication, selection of the personnel who have similar mindset, modification of work culture toward agile method, conquering the different time zones, trust between team members, and appropriate management of tacit knowledge.

**Proposition**

**The difference of team performance**

When proposing the result of the simulation, it is possible to compare performance of two different groups, the collocators and remote locators. Nan et al. (2008), a study based on Agent-based simulation for the effect of the physical proximity and computer mediated communication (CMC), suggested two competing hypotheses and tested them. The first, based on social identity theory, is the outperforming of collocators against remote locators. The reasoning of this hypothesis is that the members in a group will discriminate against the people who are outside of the group by maximizing the differences between allocations to in-group and out-group members (Jost and Azzi 1996). Another hypothesis, based on the weak tie theory, is the outperforming of remote locators against collocators. According to the theory of weak tie relationship, the relationship with relative stranger will be more effective in obtaining some resources such as job opportunities (Granovetter 1973, 1983). When comparing the situations of teams, Agile team and the general team tested in Nan et al. (2008) are similar in that the team members in both types of teams are seeking the resources for their goal achievement, and the physical proximity and CMC play their roles as obstacle or bridge respectively for/against team members’ resource seeking. Hence, we propose two competing propositions:

**Proposition 1a**: Collocated members of the agile development team will show higher performance than remote Agile team members.

**Proposition 1b**: Remote team members of the agile development team will show higher performance than collocated Agile team members.

In other hands, type of task and communication content in Agile development team can be different from those of general teams. When considering the work performed in Agile development team, the communication content is mostly about programming. The case in Sepulveda (2003) represents these differences. Agile team members use screen sharing application to share the screen with the members in remote site and use headset to talk comfortably. Also, software is defined by the specification, a kind of formal language to be used to define the functionality of software, and this specification is devised to support more clear communication.

Hence, we need to consider the communication problem of Agile development team in the perspective of media richness theory. According to the theory, face-to-face is the richest medium and the document is the poorest media. Agile manifesto and other agile practices also have emphasis on face-to-face interaction, running source code, and minimum documents (Nerur and Balijepally 2007, Lee and Xia 2010, Bose 2008), and Agile teams’ communication methods already have certain level of richness due to
the characteristics of their task. Thus, we need to analyze the effect of richness of communication method in the perspective of the media richness theory. We propose the following hypothesis:

**Proposition 2:** The level of media richness will affect (or determine) the difference of remote agile development team performance.

**Methods**

**An Agent-based Model of Distributed Agile Team**

**Behavioral Rules**

In prior studies, the similarity between Agile development and CAS are addressed and suggested that Agile methodology reflects the characteristics of CAS (Nerur et al. 2005; Meso and Jain 2006). Augustine et al. (2005) suggested APM (Agile Project Management) by using the analogy between CAS and Agile development. In Meso and Jain (2006), they also suggested that many attributes on Agile development can be compared to the attributes on CAS and suggested a mapping table. They showed mapping from seven agile practices to CAS principles in three dimensions, product dimension (artifact), process dimension (Development), and People Dimension (software team).

In this paper we propose using the Agent-based Model to test the performance differences between collocation team and remote team in Agile development setting.

As mentioned above, because of the similarity of fundamental characteristics of Agile and CAS, the Agile-based Model is superior way of modeling the remote communication process among the members of an Agile development team. By using Agent-based model, a real world environment and phenomena can be induced and particular behavior can be manipulated by including/excluding the related characteristics (Nan et al. 2008). Hence, computer model can include the characteristics that are the interest of simulation/experiment (Nan et al. 2008).

Nan et al. (2008) experimented the communication delay in remote team using agent based modeling. The model in Nan et al. (2008) is well simplified and concentrated on testing the communication delay without any confusing effect in team environment, and it has similar behavioral rules with those of Agile team. First, Favoritism Rule can also be applied to Agile team because Agile team members have the tendency of interacting with collocated team members (Sepulveda 2003). Second, it is well known that there is a communication delay with the team members who are in remote site, not only in Agile team but also in general teams (Williams 1977; Fowler and Wackerbarth 1980; Daft and Lengel 1986; Sproull and Kiesler 1986; Dennis and Kinley 1998; Burke and Chidambaram 1999 as cited in Nan et al. 2008). Thus, Delay Rule is also applicable to Agile team. Third, Perceived market Rule can be applied in the form of concentration of interaction to the Agile team. If there are several team members who can provide supports to other team members (e.g. mentor or co-ordinator in Hoda et al. (2003)), they will be busier than other team members to deal with other team members' issues as iterations are repeated, and, as a result, team members have to pay higher cost (e.g. effort to make appointment, waiting time). Fourth, Relationship memory Rule is also applied to Agile team in the perspective of Transaction cost theory because most of the people have a tendency to avoid the cost for searching new resources.

Hence, we adopt four behavioral rules from Nan et al. (2008) as a base of our simulation. The behavioral rules adopted from Nan et al. (2008) are presented in Table 1.

Based on the model we add two new rules related to agile development: the "Personal difference rule" and "Multiple expert rule".

The "Personal difference rule" represents individual differences of agile team members. While the differences vary in many attributes, programming and industry skill/knowledge and the degree of motivation toward Agile developments can affect communication between Agile team members. Lee and Xia (2010) and Balijepally et al. (2009) suggested that the team performance is proportional to the programming skill of team members. In addition, the programming and industry skill/knowledge will affect the decision making among team members. If a team member has enough skill or knowledge on the task he has to complete, he does not need to interact with other team members who has expertise. In addition, the interaction between team members will make the skill/knowledge of the task of the team
member improve (Meso and Jain 2006). Hence, as the result of interaction, one member's skill/knowledge level will increase. At the same time, by this growth of skill/knowledge, the interaction between team members will diminish as iteration proceeds during Agile development project. As a result, the overall cost of the team for interaction will be reduced. Hence, the skill/knowledge on the task is important factor in Agile development team.

Another personal difference is the degree of motivation toward Agile methodology. The more a member of team is motivated to agile development, the more the individual's action is in accordance with the suggestion of Agile methodology (Augustine et al. 2005). In a situation in which interaction is more helpful than working alone, the team member who is more motivated will conduct more interaction than less motivated team members. According to the personal difference rule, the members who have different level of motivation toward Agile development will act differently. Personal difference rule gives an agent different probability of performing interaction with other related agents. The probability will be calculated by the skill/knowledge level of an agent and the degree of motivation to Agile methodology. If an agent has enough skill, it will try to solve the problem (in this paper it is represented as the action of producing or purchasing shapes) by itself to save the interaction cost. If an agent doesn’t have enough skill and is less motivated under a certain threshold, the agent will be charged $1 as a penalty, which means the use of its own resource (time/efforts) to solve a problem by itself as initial trial.

<table>
<thead>
<tr>
<th>Behavioral Rule</th>
<th>Agents affected</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Favoritism Rule</td>
<td>Collocated Agents</td>
<td>The favoritism rule gives the collocated agents a bias toward other agents in the same location. Collocated agents prioritize incoming requests from other collocated agents, and preferentially send requests to other collocated agents. Favoritism is not operative for remote agents. Therefore, remote agents respond to requests on a first-come-first serve basis. Meanwhile, they randomly choose one between two specialty producers to request a certain shape.</td>
</tr>
<tr>
<td>Delay Rule</td>
<td>Remote agents and collocated agents who communicate with remote agents</td>
<td>All requests to and from remote agents take one extra time period to arrive at their destination. Agents cannot take any other actions while their messages are in transit.</td>
</tr>
<tr>
<td>Perceived market Rule</td>
<td>Both collocated and remote agents</td>
<td>An agent’s initial selling or buying price is the average of the costs for producing the specialty shape and the non-specialty shapes. During the simulation, agents increase their desired selling price by $1 each time they receive more requests for their specialty shapes than their production limit. Meanwhile, agents increase their desired purchasing price by $1 each time a request is denied by a seller. Once an agent receives a buying price that is higher than its desired selling price, the agent delivers the shapes.</td>
</tr>
<tr>
<td>Relationship memory Rule</td>
<td>Both collocated and remote agents</td>
<td>Both the purchasing and the selling agents increase their relationship evaluation scores of each other by one after each transaction of shapes. Agents prefer to exchange shapes with those who have earned higher relationship scores in their memory.</td>
</tr>
</tbody>
</table>

Table 2. Behavioral rules in the multi-agent model of Nan et al. (2008)

The "Multiple experts rule" is about the number of experts in the team. In traditional plan-driven project team, different roles are allocated to different people. But, Agile methodology emphasizes the redundancy of functions: “The redundancy of functions within parts, coupled with the capacity to learn,
provides for flexibility and responsiveness—characteristics critical for survival in a complex and turbulent world” (Nerur and Balijepally 2007). Although the agents in the fundamental model of this research from Nan et al. (2008) can produce all the shapes in the model, they have specialty only on a particular shape (product in a low cost). This setting is more similar to the traditional development methodology. Hence, the multiple specialties per an agent will be allocated in the present study.

To consider the effect of media richness between two agents communicating each other, “Media Richness Rule” is considered. The time required for understanding the exact specification of the software that will be developed or the time for learning the related knowledge for software development will delay the entire delivery time of the software, and the rate of delay will be dependent on the degree of media richness. Hence, for all requests to and from remote agents, additional time will be added to the delivery time to apply the effect of the degree of media richness. The amount of additional time will be decided by the degree of media richness between two agents that trade shapes.

<table>
<thead>
<tr>
<th>Behavioral Rule</th>
<th>Agents affected</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal difference Rule</td>
<td>Collocated Agents and Remote agents who are producing the shape related to purchase</td>
<td>Personal difference rule gives an agent different probability of performing interaction with other related agents. The probability will be calculated by the skill/knowledge level of an agent and the degree of motivation to Agile methodology. After the interaction, the cost of producing corresponding shape will decrease by $1 because the member’s skill on producing the shape will increase. If an agent has enough skill, it will try to produce shape by itself to save the interaction cost. If an agent is less motivated under a certain threshold, it will be charged $1 as a penalty, which means the use of its own resource (time/efforts) to solve a problem by itself as initial trial.</td>
</tr>
<tr>
<td>Multiple Experts Rule</td>
<td>Both collocated and remote agents</td>
<td>A model has more than one expert agent per skill. For example, one agent can produce all the 5 shapes, but some agent can produce more than 1 shape at lower cost than other agents. Agent will look for an expert who is not working for other members among multiple experts.</td>
</tr>
<tr>
<td>Media Richness Rule</td>
<td>Remote agents and collocated agents who communicate with remote agents</td>
<td>For all requests to and from remote agents, additional time will be added to the delivery time to apply the effect of the degree of media richness. The amount of additional time will be decided by the degree of media richness between two agents that trade shapes.</td>
</tr>
</tbody>
</table>

**Table 3. Added behavioral rules**

**Action of individual agent**

As the behavioral rules increase, the action of individual agent also changes. The changes will happen on “decision on which shape to buy” and “choice of agent to contact”. Once, agent is ready for buying, the agent will ask himself “When considering my skill/knowledge level, can I solve this problem by myself?” If the answer is “Yes” then the agent will stop and try to solve it by itself. But if the answer is “No”, then the level of motivation toward Agile methodology of the agent will be calculated. If the agent is motivated more than a threshold, the agent will not receive any penalty because the agent will interact more actively, otherwise the agent will be charged $1 as penalty which means the use of its own resource (time/efforts) to solve a problem by itself. Another change is about the agent choice process. In fundamental model, an agent searches other agents by following the collocation rule and favoritism rule. But after the change, the agent will search for the available agent based on the collocation rule, favoritism rule, and random choice of among multiple experts. Due to the adding of the Multiple Expert rule, agent might find multiple available experts. If the condition of collocation and favoritism are same, an agent will randomly choose
one expert to contact. Figure 2 illustrates how the behavioral rules work together in the multi-agent experimentation. The shaded parts of the rules are the new rules that have been added.

![Figure 2. The diagram of rules](image)

**Research plan**

**Experiment**

In this study, research on the distributed agile method is performed using agent based model. The advantage of Agent-based model is that the simulator has the control of every aspect in the model (Sycara and Lewis 2008). Because the aim of this study is to measure the team performance difference in different location and communication media, the Agent-based Model of remote Agile team is proposed.

To test the hypothesis, multiple runs of simulation will be conducted. During the simulation various types of media richness variables will be set by changing the communication delay variable in the Agent-based Model. Thus, experiment setting will be $2 \times n$ as shown in Table 4.

The performance of each agent will be measured by (1) each agent’s accumulated profit from selling shapes and filling orders and (2) quantity of shapes purchased. These two measurements are adopted from Nan et al. (2008). In Nan et al. (2008), authors used three measurements, which are the quantity of shapes purchased, purchasing and selling prices, and the accumulated profit from selling shapes and filling orders. However, the measurement for the success in resource sharing, which is the purchasing and selling prices, will not be used for this study because interactive and cooperative way of work paradigm such as the pair programming is more encouraged in Agile development environment than the success in resource sharing (Nerur et al. 2005). Since the on-time delivery of completed software is the measurement of the performance of Agile team (Lee et al. 2010), we will use measurement (1) as the
measurement of overall performance of Agile development team. Through the statistical analysis of the result, we will test whether the treatments of this experiment have the significant differences on the performance of Agile development team and will conclude which propositions are supported. Apart from the measurement (1), the measurement (2) also will be tested through the same procedure to see the difference of the degree of interactions among Agile team members because the measurement (2) represents the Agile team members’ effectiveness in obtaining scarce resources.

<table>
<thead>
<tr>
<th>Media Richness Level</th>
<th>Collocated Agile Team</th>
<th>Remote Agile Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Level n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4. Experiment Setting**

In addition, we may simulate various types of what-if questions by changing the kinds of variable and functions with this Agent-based Model. One expected simulation is comparison of team performance in large and small scale team by changing the number of agents and by adding the concept of clarity of information about the location of experts. For example, by increasing the number of agents in the team, the Agile team will have more chances of interaction and more abundant expert pool, but also will have less clear information about the location of experts for their task.

**Analysis**

After multiple runs of simulation, statistical analysis will be conducted to test the hypotheses. According to Nan et al. (2008), although analysis of variance (ANOVA) is often performed to analyze repeated measures, mixed model (also called hierarchical linear models, nested models, random effects models or generalized linear mixed models) is more appropriate because the interdependence and the nested structure among observations violate the fundamental assumptions of ANOVA. Since the simulation in this paper has similar characteristics to that of Nan et al. (2008), we also will use mixed model in our analysis. After the statistical analysis, result of analysis and statistical meaning will be discussed.

**Conclusion**

During the past decade, Agile methodology has been used as a new approach for software development to deal with complex and unstable requirements of business. It is a useful tool, but still has questions about the control of Agile project due to the uncertainty caused by its CAS-related characteristics. Although many previous studies have found the elements of Agile methodology, they didn’t provide any holistic view of this complex adaptive system.

This study is conducted to provide the macroscopic view for the mechanism of Agile team. In this study, Agent-based Model of Agile team is build and simulated, and the answers for our research questions are found. As the first contribution, we acquired the knowledge about the relationship between distributed Agile team and the physical location and the relationship between distributed Agile team and the degree of media richness. In addition, as a second contribution, we established Agent-based Model of Agile team. The advantage of this model is we have a tool to carry out the various types of simulation for Agile team.
References


