Reducing Complexity Faced By Large Organizations Using Distributed Scrum in Agile Development

Monika Agarwal
Department of Computer Science
Amity University, Noida, India

Rana Majumdar
Department of Information Technology
Amity University, Noida, India

Abstract— A complexity faced by almost all large organizations is distribution of teams (many teams working in different locations) and dispersed teams (team members within a team are dispersed to different locations) When organizations begin to use Scrum, they often run into the difficulties inherent to distributed and dispersed teams Because Scrum uses frequent inspection and adaption of transparent artifacts to control risk and create predictability, distribution and dispersion make these techniques difficult. For instance, if I have never worked closely with someone on my team from Asia and I am in Boston, how do I interpret what he or she means in the Daily Scrum? If we have many teams scattered throughout the world, what techniques can we use to resolve our dependencies and frequently integrate our work, so that transparency is retained? There are no optimal solutions; only best possible solutions. The thing doesn't get easier when Scrum is applied. As the number of teams and locations increased build time grew longer and longer. We also have to figure out how to run the sprint meetings with teams distributed across several times zones. When agile was introduced in the mid 90's, collocated development teams were common (that is, the whole project team was in one location).So the daily Scrum meetings could be conducted in the same place at the same time. Nowadays, a distributed development team is the norm as companies embrace global sourcing. A project even could have team members located in different continents. The challenge is how to apply the agile principles that were established in a collocated environment to a distributed development environment. Current paper is on how to use Scrum in very large, heterogeneous, globally distributed organizations. It also provides an awareness of the problems that are likely to arise on a distributed Scrum project. It excels at providing practical tips to avoid and overcome these issues [6].

Keywords— Agile, Agile Principle, Distributed Development Team, Scrum, Sprint, Sprint Meeting.

I. Introduction

Distributed development is a fact of life for many teams. Unfortunately most agile methodologies or approaches assume that the team is located in a single team room. Until recently there has been little guidance about how to apply these approaches with a geographically dispersed team. In The 2008 State of Agile Development survey, conducted by Version One, [2] 57% of respondents stated that their teams were distributed. Furthermore 41% of respondents said that they were currently using, or plan to combine, agile with outsourced development. When faced with these sorts of numbers it seems that the agile practice of placing the entire team in a single room is at odds with what's actually going on within a large part of the software development community [1].

II. Why Distributed teams?

From the above discussion we reach to a point that the distributed teams are the demand of current scenario but why? Here are some reasons:

A. Globally distributed teams to reduce cost

A software development organization may outsource a particular function to another company in another country, or the software development organization may be hiring their own workers in an emerging country. With a well thought-out plan to best leverage the talent in multiple countries, it can be less expensive to develop a product. Working with distributed teams where the talent is available to do the work can sometimes reduce labor and business operations costs [6].

B. Reaching Market More Quickly with the "Follow the Sun" Model

The team with members in the U.S. and China essentially has the possibility of working on a project 24 hours a day and making significantly more progress than a team working on a standard (8-hour) workday. By working with distributed teams, it is possible for companies to work a full 24-hour day and get their product to market more quickly [6].

C. Distributed Teams Expand Access to New Markets

Most of foreign affiliate sales are local sales, and local sales in emerging markets increased between 1994 and 2002 (Borga 2005) [7]. To parent companies, this underscores the importance of investing in foreign affiliates.

The world's top 100 non-financial transnational corporations, ranked by foreign assets in 2003, showed the UK, Netherlands, Japan, France, Spain, Germany, Switzerland, the United States, Malaysia, Australia, Norway, the Republic of Korea, and Ireland are all home to leading corporations that invest heavily outside their home country. Almost percent of the employees in the top 20 transnational corporations work outside of the home country (United Nations).

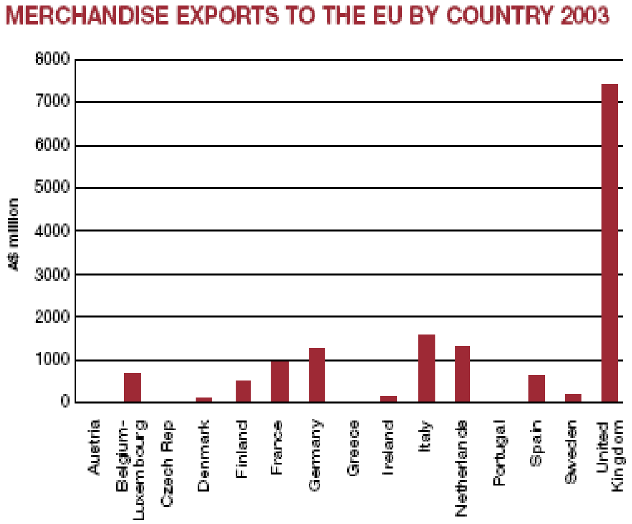

Figure 1: ABS data on the DFAT STARS Database

IBM continues to drive growth in emerging markets, which accounted for 18 percent of IBM's overall geographic revenue in the second quarter of 2008, and represented a growth rate of 21 percent year-to-year (Azzi 2008)[8]. The Brazil, Russia, India, and China subset grew 31 percent in the second quarter of 2008, led by strong growth in India. IBM continues to grow, nurture, and develop software development team members in these markets. In January 2009, the IBM Scrum.

Community included software development team members from more than 30 different countries as a result.

D. Acquisitions

Another trend that increases team distribution is market consolidation, which results in an increase in acquisitions. These acquisitions are likely to result in distributed teams as the combined companies begin working together to integrate their products. Within IBM, Cognos® is a good example of this. Cognos bought several companies with development teams in multiple locations before IBM bought them. After joining IBM, the different Cognos branches began working with IBM teams in even more distributed locations.

E. Expanding for Innovation and Thought Leadership

The search for knowledge, intellectual property, and innovation are also responsible for team distribution. Studies have found that multinational companies produce more ideas than their purely domestic counterparts (Criscuolo 2005) [9]. Having more researchers and access to a larger "worldwide pool of information" fosters the generation of more ideas. In a knowledge-based economy, innovation and thought-leadership are critical to the success of a business [10]. Hiring software development team members in other countries as part of the search for innovation naturally increases distribution of teams. [6]

III. Challenges Of Scrum In Distributed teams

As described in the Agile Manifesto, Agile power resides under the team interaction on daily basis. Agile Methodologies such as extreme programming scrum, lean development, Feature-Driven Development, Adaptive Software Development, Dynamic Systems Development Method, Crystal – A Family of methodologies should follow the same principle but in distributed environment, it is very difficult to follow. It is about instant decisions and collaborative work rather than extensive documentation and cumbersome processes. One of the key elements of this collaboration is that the entire project team sits ideally in the same room: the business owner, the frontend developers, the Scrum master, the database developer, the system administrator, the QA developer. If your company has outsourced its development team and is thinking about using Scrum to develop software, you are probably trying to find out a process to overcome the distance factor. Here is a non-exhaustive list of what challenges I, as a Scrum Master, have experienced so far in the use of Scrum within distributed teams [3].

A. Communicating with Distributed Team Members

Distributed teams experience the same problems as collocated teams, plus they have some extra ones. Many of the problems are communication issues at their core because of the added difficulties posed by distance. Even when team members are on the same continent or in the same region, not working face to face can complicate communication. The complexity increases with distance as time zones, language barriers, and cultural differences get in the way.

B. Time Zones and Working Hours

Misunderstandings about meeting dates and times occur more often with a distributed team working in different time zones than with a collocated team.

C. Cultural Differences

Cultural differences can also impact the effectiveness of a team's communication and collaboration. These differences become more obvious when a team works closely together as part of a Scrum team throughout a Sprint. As an example, in some cultures, it is inappropriate for someone to say they do not understand the speaker. The primary symptom of this is finding that some team members say "yes "without ever challenging or questioning statements. This can be problematic for a distributed team where communication in general is much more challenging. Identifying points that need clarification is especially critical to align the team's efforts with the expectations of the Product Owner.

D. Language Differences

A team distributed across regions or countries may also face language differences that will impact their ability to communicate effectively. Language challenges appear in several forms. Some or all distributed team members may not have the same first language, some members may not speak the language used by the team, or team members may not be able to easily understand one another because of accents. A Scrum environment where team members communicate often will highlight communication challenges and make them more obvious.

E. Tools

Locations with a less-advanced communication infrastructure than others can pose problems for the distributed team. To increase productivity of distributions teams, issues with communications tools must be identified and addressed. Do not assume the tools and environment available to team members in one location are the same for team members in another location.

F. Schedule Differences

Scheduling team actions and tasks becomes more challenging with increasing levels of distribution. Within IBM, each site has a local holiday schedule [10]. Although these are consistent for sites within a given country, holidays across countries can vary. The Chinese celebrate their New Year for a week, beginning between January 21 and February 20. The U.S. teams typically take vacations during the last two weeks of December. Some in India celebrate the five-day festival of Diwali beginning on the new moon that occurs between October 13 and November 14.

G. Team Dynamics

Distance can make some team members feel isolated, especially if they have a specialized skill that is used only at certain times during the development effort for each work item. If the team only engages the writers on their team after a feature has been demonstrated to stakeholders, writers can feel they are not part of the team. Team members that are on a waterfall team may not engage with one another for long periods of time. Not having regular contact with other members of the team can weaken the social skills of remote team members. They may feel less comfortable with dealing with one another. [6]

H. Building Trust

The other key enabler – or constraint – for distributed projects is how much trust there is between the Product Owner and the team. Inevitably, in the course of day-to-day cooperation, there will be bumps in the road Miscommunication will happen, misunderstandings will occur, mistakes will be made, and myriad other problems will come up. It is not uncommon to find distributed projects where the Product Owner is utterly convinced that the team is incompetent, and the team is utterly convinced that the Product Owner is incompetent. With limited information about the other person, we often tend to fill the gaps with fears rather than facts; when someone does the wrong thing, we are apt to take it as evidence that they do not know what they are doing – which is what we fear most – rather than other possible explanations (such as: they did not fully understand what was expected of them and were afraid to ask for clarification). [4]

IV. Ways To Reduce The Complexity Faced By Scrum

A. Enabling Communication

To ensure an effective communication between the product owner and team we have some ways as-
First, there are practical considerations. The various modes of communication – email, telephone, face-to-face conversation can be placed on a "richness" scale, which looks something like this:

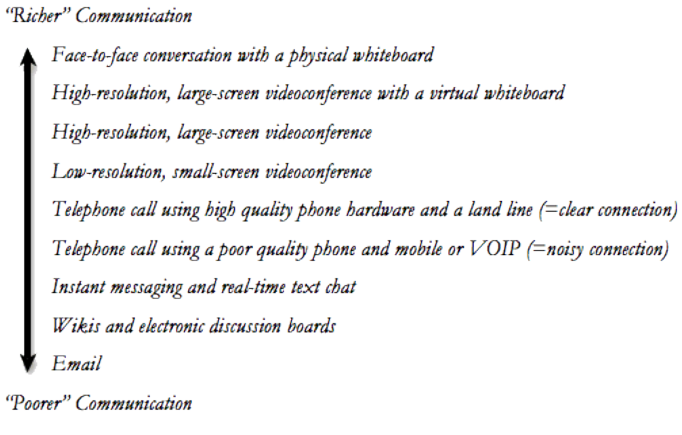
"Richer" Communication
Face-to-face conversation with a physical whiteboard
High-resolution, large-screen videoconference with a virtual whiteboard
High-resolution, large-screen videoconference
Low-resolution, small-screen videoconference
Telephone call using high quality phone hardware and a land line (=clear connection)
Telephone call using a poor quality phone and mobile or VOIP (=noisy connection)
Instant messaging and real-time text chat
Wikis and electronic discussion boards
Email
"Poorer" Communication

Figure 2: The Richness Scale

By and large, the higher up this scale you are, the richer and easier the communication, the more natural the interaction and the more immediate and faithful the understanding between people. Email is, unfortunately, the go-to mode of communication between most distributed Product Owners and teams, and this is a mixed blessing. Its great strength is that it is not dependent on both parties being present simultaneously, and it preserves a record of the discussion that can be referenced later. The big disadvantage of email is that it is often much more time and effort-intensive. A discussion that might otherwise require a single, five-minute telephone chat could easily turn into 10 back-and-forth emails, each cc to other people (thus consuming their time and attention, if even just to hit the delete key). Email conversations also breed misunderstanding, and as a result, unnecessary or unintended emotionality; without the subtle cues of voice intonation and facial expression, one can easily misunderstand the mood, tone, and intent of the writer.

In the team work area, there should be a high-quality speakerphone (for example, a Polycom Sound Station) with the speed-dial buttons programmed to the Product Owner's desk and mobile phone numbers (preceded by any long-distance "unlocking" codes), plus a sticker attached to the phone with the acceptable local hours to call (or clocks with the different location times).

In addition, each team-member's desk phone or VOIP application (and if possible, mobile phone as well) should also have the Product Owner's telephone numbers programmed on speed dial. Enabling easier telephone communication is an important step, but it is not enough. All of the key Scrum meetings – Sprint Planning, Product Backlog Grooming, Sprint Review, and Sprint Retrospective – should be conducted visually. The problem with audio-only meetings is many. One misses out on facial expressions and body language entirely. It can be unclear which voice belongs to which person. The natural "flow" and cadence of a conversation is often missing; there are either unintentional interruptions, or people are afraid to speak up for fear of interrupting. If participants have unfamiliar accents, it is harder to understand them without a view of their face as they speak. However, the most significant dysfunctions of voice-only calls is people "multitasking" during the call; without a visual on what they are doing, people will often find checking email or surfing the Internet irresistible, and only pay partial attention to what is being discussed. Participants are effectively only "half-there." Some companies have invested in sophisticated videoconference equipment, but teams may find it complex and cumbersome to operate, or the conference room where it is located is often booked. It may be more effective to provide the team with an improvised solution as follows:

Video: Skype with a wide-angle high-resolution webcam. (It is important to use a wide-angle webcam – this gives a wider field of view, enabling more people to be seen on-camera)

Audio: High-quality conference phone connected via a land-line, with multiple extension microphones for the table. (In some cases doing the audio via Skype is sufficient, but generally a high-quality conference phone on a land-line will produce much better fidelity.) Ideally, the above equipment should be set up and ready to use at any time in the team room, and this should be replicated at the Product Owner's side. Finally, for the key Scrum Meetings – Sprint Planning, Product Backlog Grooming, Sprint Review, and Sprint Retrospective – it's helpful to have simultaneous videoconferencing and white boarding capability. The following diagram shows a conference room with an ad-hoc setup for doing this, with two projectors side-by-side (one projector displaying the Skype video feed from the other location, and the second projector displaying the shared desktop or virtual whiteboard), plus a high-quality conference phone on a land-line.

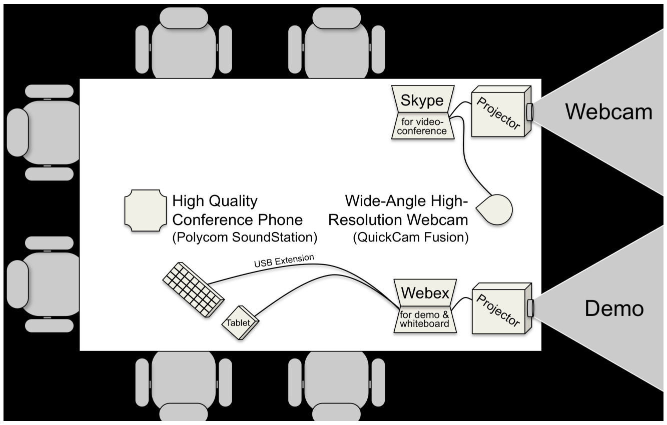

Figure 3: A conference room with an ad-hoc setup

B. *Building Trust*

The only way to reduce the risk of these misapprehensions taking hold is by building a foundation of trust between the Product Owner and the team. This begins with a human relationship between the two. One of the most critical steps for the success of a distributed Scrum project is for the Product Owner and team to come together in person at the beginning and spend quality time sharing key project information and building a relationship with each other. To establish a foundation of trust at the beginning of a long-distance working relationship, it can be very helpful to have an open and direct conversation about what everyone is committing to, and what each expects the other to do. This could simply take the form of a conversation, or it could come in the form of a "working agreement" between the team and the Product Owner.

C. Distributed Scrum Practices

In a distributed environment, all the standard practices of Scrum – the roles, meetings, and artifacts – are present. However, it may be necessary to adjust how those practices are implemented, to overcome differences in time zone and geographic location.

1) Sprint: There is no "best" Sprint length to use, either in a co-located or a distributed environment. Longer Sprints (3 or 4 weeks) enable teams to produce larger increments of functionality each Sprint, and Sprint Planning and Review / Retrospective (which typically involve early morning or evening meetings for everyone involved) occur less frequently. Unfortunately, both of these benefits can create other drawbacks. Because of the communication problems that flow from having the participants in different locations, it is far more common to discover misunderstood requirements when we reach the Sprint Review. In a 4-week Sprint, it is possible that twice as much of the "wrong" functionality will have been built than would have been built in a 2-week Sprint. Additionally, a 4-week Sprint offers half the frequency of inspect-and-adapt cycles for the team's practices, so many teams find they have fewer opportunities to surface and address dysfunctions. One solution is to start with 2-week Sprints, and focus initially on mastering the ability to deliver increments of potentially shippable product (possibly very small ones) by the end of a Sprint. A number of Sprints' worth of inspect-and-adapt may be required for the team to achieve this, but once they have succeeded, they can shift to a longer Sprint length, and be able to deliver larger, more satisfying increments of functionality.

2) Stakeholders: Stakeholders are individuals who can affect or be affected by the project either positively or negatively. They could include sponsors for the project, customers interested in using the final product, and vendors who may distribute the product. A key strength of the Agile methodology lies in its ability to involve the customer as a relevant stakeholder and ensure that the expected level of stakeholder involvement occurs. This facilitates customer involvement with each deliverable. If the interaction is not up to expectations, then there is a provision to communicate this to the appropriate level, individual, or group in the organization. This allows quick corrective action as the success of the project depends on stakeholder participation at all levels.

A stakeholder is responsible for these tasks:

- Helping the product owner define the goal and vision of the project.
- Attending planning meetings and helping team members with prioritizing components on a regular basis.
- Optionally attending daily team meetings but not participating in the proceedings.
- And, attending review meetings to assess regular progress.

3) The Product Owner: The Product Owner is responsible for gathering inputs about a product from the customer and translating the requirements into the product vision for the team and stakeholders. The Product Owner can either be an individual or an organization. In some cases, the Product Owner is the customer. In other cases, where there are multiple customers with a variety of requirements, the Product Owner role maps to the Product Manager or Product Marketing Manager. The Product Owner represents all the stakeholders and translates their requirements into a product road map. He or she then focuses on the business side of the product and helps to divide the project into measurable project deliverables prioritized by business value.

The Product Owner is responsible for these tasks:

- Consolidating all client requirements into a prioritized set of deliverables.
- Defining the features of the product based on market value.

- Managing project ROI and risk.
- Adjusting features and their priority on a regular basis.
- Participating actively in all planning and review meetings of the project.
- Providing answers to questions raised by project teams from time to time.
- Determining the release plan for deliverables and communicating it to all stakeholders.
- And, accepting or rejecting work results.

4) The Scrum Master: The role of the Scrum Master becomes even more critical in a distributed project, because the "dysfunctions of distance" – for example, the difficulty of communication -- require an even more active and tenacious commitment to openness and inspect-and-adapt. Day-to-day, distributed projects also tend to have a greater-than-usual load of impediments, obstacles, and disruptions that will require the Scrum Master's attention and effort. If the Product Owner is in one location and the team is in the other, the Scrum Master should be located where the team is. While an onshore Scrum Master may be able to help an offshore team with some types of issues, he or she will unfortunately be entirely absent from the realities of the team's day-to-day work life, and thus they will be far less useful to the team. The critical Scrum Master Duties of coaching the team, helping remove impediments, and protecting the team from disruption will essentially be absent if the Scrum Master is located far from the team. If the team itself is divided between multiple locations, there should be a primary Scrum Master designated for the team overall, but it will probably be helpful for each location to have a team member playing the role of "local" Scrum Master during that location's working hours.

   These are some of the tasks of the Scrum Master:

- Helping the team by trying to resolve issues or blocks that the team faces.
- Improving the productivity of the team and keeping it effective.
- Enabling close cooperation among various roles and functions.
- Keeping track of training requirements for the team to avoid delay in deliverables.
- Protecting the team from outside interference or disruption.
- Protecting the interest of the Product Owner and maximizing the ROI.
- Organizing and facilitating Scrum-related activities and meetings.
- And, ensuring that the team follows Scrum standards.

5) The Team: When the team itself is split between multiple locations – for example, several team members are located in China, and several team members are located in the US – the challenges of development are often multiplied further. The level of coordination, cooperation, and teamwork that is necessary to deliver working software every 4 weeks or less is even more demanding.

   The Scrum Team is responsible for these tasks:

- Selecting the goal and specifying work results.
- Organizing the team and work.
- Demonstrating work results to the Product Owner.
- Providing the Product Owner with inputs and ideas about the final product.
- And, working within project guidelines to reach the goal.

Team members will typically need to spend periods of time working side-by-side with each other, especially at the beginning of the project. An excellent practice is for the team to be collocated for the entire first Sprint of the project, and ideally the first two or three Sprints.

This allows the individual developers to build working relationships with each other, as well as trust and visibility into each others' skills, strengths and weaknesses. In addition, the team will develop a set of working agreements and set standards for their "definition of done," quality, coding conventions and other development practices, tools, escalation, overlapping work hours, and other necessities. In addition to collocation of the team for the first Sprint or more, the distributed teams that succeed with Scrum typically have some sort of "ambassadorship" practice, where team-members are constantly traveling to the other location for periods of time working side-by-side with their distant colleagues. When these "ambassadors" return home, they bring with them knowledge and values that will inform the work of their local colleagues, and they will also be able to function as points of contact for their remote colleagues when issues arise. This ambassadorship is ideally a continuous practice, with one team member placed at the other location at all times, in rotation. The immediate objection to this constant travel is “it will cost money and time!” The simple response: Yes, but it is cheaper than the alternative, which is getting much less business value from the project. Without face-to-face contact and high-quality working relationships, the team will produce less software, lower quality software, or functionality that’s less right for the customer needs – or all three. A team of 6 developers with a generous travel budget will probably produce much more business value than a team of 7 developers with no travel budget. A common dysfunction when the teams itself is split between two locations and are not properly bonded is that “one team” actually operates as two teams. Team-members may form “cliques” by location, and miscommunication and in the lack of coordination between the locations may give rise to mistrust and conflict. It is also possible that if a portion of the team is located onshore, they will be in closer communication with the Product Owner, and as a result they will have an information advantage over the offshore developers. While one would hope that this could benefit the entire team, it can sometimes do the opposite, driving a wedge between the two groups of developers, with the offshore group being seen by onshore as clueless and always a step behind, and the onshore group being seen by offshore as hoarding knowledge and looking out only for themselves. Rather than trying to work as a single team, it may be more effective to form into separate Scrum teams, one per location, and as loosely coupled with each other as possible. Each team should be fully cross-functional, and should be responsible for producing entire pieces of functionality, not simply doing a particular activity (coding, testing, etc.).[4]

V. CONCLUSION

Some traditional activity needs to be changed for moving from Distributed Traditional approach to distributed agile Approach. Many tool such as Microsoft Share Point, Rally or Version 1. Visual studio SourceSafe are available to keep the control of project development and team’s status in distributed environment. Agile techniques actually help you to address and mitigate the usual challenges of a distributed / off-shored project: lack of visibility on project status, delay in feedback cycle, loss of business and technical contexts, decrease in communication bandwidth, higher documentation overhead, and mistrust.Short iterations, with a product demo and retrospective at the end, increase visibility of the project status and provide instant feedback as well as an opportunity for process adjustment. Customer involvement facilitates both a shared understanding of business context and communication between business people and the development teams.

Frequent team meetings help to build trust and improve communication on all levels and across different groups. Continuous test and integration cycles tell us where we are in the project. Functional Test Driven Development helps in removing ambiguity from requirements and clearly communicating them to distributed teams. Before adoption and making the better use of agile methodology, the organization has to understand what challenges can come in distributed market and how that might solve or reduce when the Team is distributed to get the best results from them.

VI. REFERENCES

[1] Distributed Agile Development at Microsoft patterns & practices Ade Miller, Microsoft patterns & practices October 2008.

[2] 3rd Annual Survey: 2008 “The State of Agile Development” http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf

[3] Barlow, Jordan B.; Justin Scott Giboney, Mark Jeffery Keith, David W. Wilson, Ryan M. Schuetzler, Paul Benjamin Lowry, Anthony Vance (2011). "Overview and Guidance on Agile Development in Large Organizations”

[4] Distributed Scrum: Agile Project Management with Outsourced Development Teams (Jeff Sutherland, Anton Viktorov, Jack Blount, Nikolai Puntikov, 40th Annual Hawaii International Conference on System Sciences (HICSS'07), 2007

[5] Principles behind the AgileManifestoagilemanifesto.org/principles.htm

[6] “A Practical Guide to Distributed Scrum”. Elizabeth Woodward, Steffen Surdek and Methew Ganis.

[7] Borga, M. (2005, September). Trends in Employment at U.S. Multinational Companies: Evidence from Firm-Level Data.

[8] Azzi, M. (2008, July 31). IBM Growth in Emerging Markets Fuels Lotus Momentum. Retrieved February 26, 2009, from Marketwire: http://www.marketwire.com/press-release/Ibm-NYSEIBM-884845.html

[9] Criscuolo, C. H. (2005). Global Engagement and the Innovation Activities of Firms. Cambridge:National Bureau of Economic Research.

[10] http://www.research.ibm.com/pdfs/scrum/Chapter1_Pearson_20090722.pdf