ETM
ENGINEERING & TECHNOLOGY MANAGEMENT

PORTLAND STATE UNIVERSITY

How Large or Virtualized Teams Affect Agile Processes

ETM-545 Team 1, Final Project

Dr. Anderson

Fall 2009

Neil Runde
Opinderjit Bhella
Dinesh Reddy
Maher Almasri
Faisal Alfayez
Srilaltha Kothapally

Abstract

The paper demonstrates the trend taking place in the software industry of organizations adopting agile processes while compensating for virtualized teams. Through research, we discovered that success in agile methodology is heavily based on face-to-face communication with small functional teams that are collocated—bigger non-collocation teams provided diminishing returns on agile philosophy. In other words, the bigger the team and the further apart you are, the more you need to start documenting and more documentation means less agile.

Keywords: agile, software development, face-to-face communication, collocated, virtual teams, SDLC

Introduction

As high-tech markets are battling a "civilized war [1]," R&D executives are seeking creativity and innovation in international design teams in hopes to access dispersed knowledge and skills [2] [3]. While these strategies are essential for business success, they do create new challenges for project managers: how to manage these "virtual teams." In parallel, many organizations are embracing adaptive or agile software methods in reaction to highly dynamic market forces and platform environment changes. This paper will demonstrate that these trends may be in conflict.

This paper is organized by defining key terms describing end points on a spectrum of software methodologies by using industry recognized characteristics. This paper will use literature review to gather data on problems that are inherent in both large and geographically dispersed teams.

Lastly, this paper will show that agile methods rely more on high-band width communications then traditional methods. The authors will then demonstrate that organizations are trending toward agile methods.

Definition of Terms

The lack of a software engineering recognized "professional" or licensing standards means the field's language has a greater range of interpretation. As such, the authors, to a degree, had to generalize on terms for the contextual orientation part of this paper.

Software Development Methods Spectrum

All software development lifecycles follow a similar pattern: problem identification; requirements generation; solution development and deployment. What distinguishes the extreme ends of the development spectrum is how much inventory of requirements the project team attempts to process through a planning, estimating, developing and testing cycle.

For a classic stage gate process, large chunks of requirements are processed through compartmentalized series of formal linear stages. The formality and compartmentalization is emendable to a functional arranged organizations or geographically dispersed teams.

When strictly followed, the entire input from one stage has to be processed and validated before the next stage can begin. There is no agreement in the literature on the definition of the stages [4]. For this paper, the authors will use the Project Management Institute definitions: Requirements, Design, Build, Inspect and Operational Transfer [5].

There are a plethora of other methods and their variants. To help describe the spectrum, see the Table 1 below which shows the methods by their agility rank which was based on their strength of constraints [6].

Agility Rank	***Method***	***Very Low***	***Low***	***Med***	***High***	***Very High***
1	Scrum	▨				
2	Adaptive SW Development (ASD)	▨	▨	▨		

3	Lean Development (LD)	▨	▨	▨		
4	Crystal	▨	▨	▨		
5	eXtreme Programming (XP)		▨	▨		
6	Dynamic Systems Development Method (DSDM)		▨	▨	▨	
7	Rational Unified Process (RUP)		▨	▨	▨	
8	Team Software Process (TSP)		▨	▨	▨	
9	Feature-Driver Development (FDD)		▨	▨	▨	
10	Capability Maturity Model Integration (CMMI)		▨	▨	▨	▨
11	Capability Maturity Model for Software (SW-CMM)				▨	▨
12	Personal Software Process (PSP)					▨
13	Clean Room/Stage Gate					▨

Table 1 Methods Ordered by Number and Strength of Constraints [6]

Stage Gate (predictive)

To articulate the linear process of the stage gate method, the authors will describe, at a summary level, what is entailed for each 'stage'. For all stages, there is a formal sign off ceremony marking the end of each stage, which includes schedule and estimate updates. A key characteristic of the stage gate model is that a stage can be concluded when the inputs of the next stage are achieved.

Business Analysis Stage

Using a *strict* stage gate model, business analysts are required to first survey the customer landscape to determine key problem sets. Once the problem set(s) reach critical mass or a specified time period has expired, the business analysis stage ends and a notion of a release begins. Senior management, generally in charge of funding projects, will decide what projects are to continue beyond this stage to transcend to the next stage [5].

Requirements Stage

Entering the requirements stage, the problem set is converted to enumerated product requirements. Once identified, they can be categorized, prioritized and roughly scheduled.

In stage gate, requirements are to be consumed at a much later date; therefore, multi-dimensional detail is necessary. Data has to be maintained in a persistent state because of this temporal issue. Requirements may utilize use-cases to describe a particular problem in common language to give specification designers an idea of the problem they are solving [5].

Design Stage

This stage specifies features in detail, including user interfaces, business rules, process diagrams, architectural diagrams and outlines of documentation [8]. Requirements are broken down into engineering language and constraints are noted. Any changes to the requirements after this stage need to negotiate through a formal change control process involving all functional teams. Because all the requirements are to be completely broken down into a specification, the change process is time consuming. For example, the team cannot rely on temporal memory so requirements are often placed in persistent storage such as a requirements or specification database [5].

Build Stage

Specifications are converted into machine code in this stage, the GUI is generated, and unit tests are run. This is the largest and most expensive stage in terms of resource usage. This stage is where everything scoped is developed. Installers, documentation, help and infrastructure systems are built to support the product. Developers attempt to manage risk by developing critical portions of the code up front. Work can be divided up by functional areas, architectural boundaries, etc, and finally brought together in an integration step [5].

Inspection Stage

Deliverables are audited for quality, completeness, etc, against the specification and reports are generated to determine possible rework [5].

Operational Transfer Stage

In the operational transfer stage, products are deployed to another entity. This process varies greatly depending on the nature of the product [5].

Authors' Empirical Observations of Stage Gate

The ceremonial characteristics of stage gate processes necessitate prescribed standards and methods. Inherent in prescription is generalization which causes poor fitting constraints on highly dynamic problems. Prescription is inflexibly and calcifies quickly. Some organizations measure the performance of their project managers based on adherence to process instead of actual results. In turn, some project managers follow process in name only which causes the calcification problem.

Adaptive (agile)

Agile was founded by a set of software engineering professionals in 2001, Kent Beck, James Grenning, et al [7]. The group was motivated by finding a lightweight alternative to the less than flexible stage gate model. They generated and published a four line manifesto [7]:

- Individuals and interactions over process and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

In support of the manifesto, they also created twelve principles, seven of which are communication oriented, therefore, relevant to this paper and listed here [7]:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software
- Welcome changing requirements, even late in development
- Agile processes harness change for the customer's competitive advantage
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale
- Business people and developers must work together daily throughout the project
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation
- Working software is the primary measure of progress

The principle addressing face-to-face conversation supports this papers contention that agile, at its most fundamental, relies on collocation for higher-frequency and high-bandwidth communication.

Agile methods are adaptive software development processes where: product requirements are fluid; planning and development activities are rhythmic; release quality is maintained and integration of components is frequent. For agile planning, planning is done in a focal-length manner where close activities are broken down to the task level and activities to be completed later are planned at a much higher granularity [8]. For agile development, user stories are gathered in small chunks and processed through to completed and shippable components. All planning and development is done in short iterations, typically 2-4 weeks.

Hypotheses

1. All projects have diminishing returns by adding personnel to increase team capacity; however, strict agile methods have a greater rate of diminishing returns than strict stage gate methods.
2. In any project, communication bandwidth is lessened with virtual teams; however, strict agile methods are impacted more negatively than strict stage gate methods.
3. The software industry is trending to adopting agile methods.

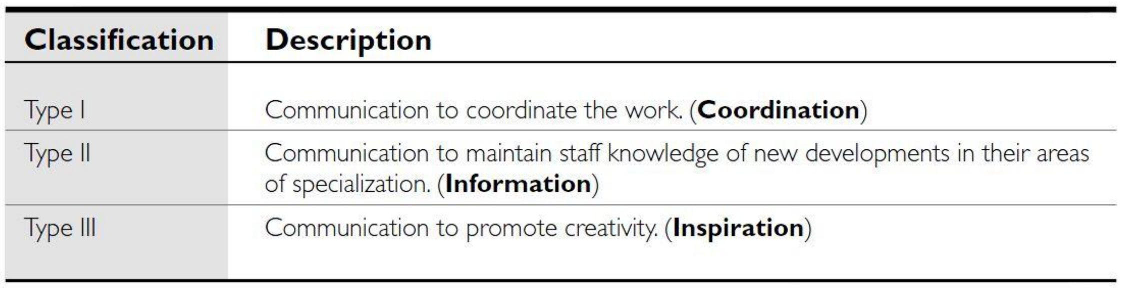
Literature Survey

This paper will use a literature survey to gather requisite data for analysis to support or refute the hypotheses. The literature origins will range from books, magazines, online articles and research papers.

We narrowed our search to issues with sub-optimally large and geographically dispersed teams that struggle to maintain necessary communication richness for project success whether agile or stage gate.

Communication Richness

The definition of communication is to transfer information from a transmitter to a receiver through a physical media. According to Allen [9], there are three classifications of communication; Coordination, Information and Inspiration (*see Table 2*). Coordination is about doing the “real” work and is fundamental to all organizations. Since employees like open and honest communication [10] [11], Information is about keeping all staff members informed of the “important” activities. And finally, Inspiration, in this type of communication creativity is considered a significant factor. Allen notes that among the three types of communication, Inspiration is considered the most unpredictable.



Classification	Description
Type I	Communication to coordinate the work. (**Coordination**)
Type II	Communication to maintain staff knowledge of new developments in their areas of specialization. (**Information**)
Type III	Communication to promote creativity. (**Inspiration**)

Table 2 Communication Classification [9]

Team Size

Teams are fuzzy things as team members can come and go throughout the project lifecycle —the boundary of team is can be unclear [12][13]. According to Katzenbach, “a team is a small number of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they are mutually accountable.” [14]

Team Size Variation over the SDLC

Due to the changing nature of skills required at different points in a project, team size is not a constant. It can be shown that resource usage throughout the software development lifecycle (SDLC) has the signature of the normal distribution as seen in figure 1 from class materials [15].

“Team Size includes employees related to all aspects of software development and delivery” [16]. This statement is meant to encompass cross functional roles such as marketing, technical writing, software developers, project managers, project leads, software quality engineers, et al [16].

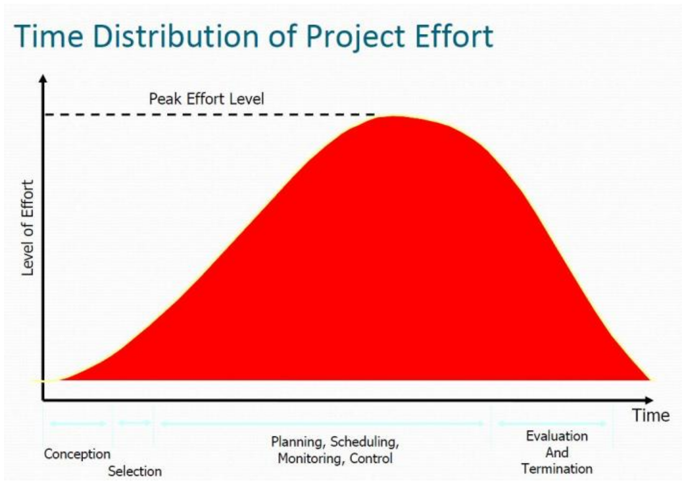


Figure 1: Time Distribution of Project Effort [15]

Affects of Team Size

According to Abrahamsson, non-agile methods are better suited for requirements that can be predefined; conversely, projects where requirements are changed frequently need adaptive methods, i.e. agile. The nature of most software projects is that requirements are fluid. The success of agile is the method's ability to deal with fluidity using iteration boundaries as requirements change points [17].

The Documentation Issue

When rapidly changing requirements are the norm on a project, keeping documentation up to date becomes a difficult process. Teams of a sub-optimal size [18] exacerbate the stale documentation problem. Increases in staff for capacity create addition communication complexity and the need for documentation [17]. In their book, Balancing Agility and Discipline, Boehm and Turner noted:

"*When agile methods employ documentation, [the] emphasize is doing the minimum essential amount. Unfortunately, most plan-driven methods suffer from a "tailoring-down" syndrome, which is sadly reinforced by most government procurement regulation. These methods are developed by experts who want them to provide users with guidance for most or all foreseeable situations." [6]*

Agile does not preclude documentation, but as stated in the manifesto the preference is: "Working software over comprehensive documentation." Documentation is also not the preferred method for team communication as also stated in the manifesto:

"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."

Agile methods require close, informal and continuous face-to-face interactions at two levels: (1) among team members; (2) between the development team and the client [19]. As team size increases, it becomes necessary to leverage technology to fill communication gaps [19]. According to Bradner et al:
"Certain types of teams may be more likely to adopt one kind of technology over another. We found that smaller teams adopted collaboration technology, while larger teams were more likely to adopt technology designed to assist in their coordination efforts." [20]
The "need" for face-to-face communication discussed above implies collocation which supports hypotheses 1 & 2:

- All projects have diminishing returns by adding personnel to increase team capacity; however, strict agile methods have a greater rate of diminishing returns than strict stage gate methods.
- In any project, communication bandwidth is lessened with virtual teams; however, strict agile methods are impacted more negatively than strict stage gate methods.

Scaling Agile

A common issue discussed at agile conferences and discussion boards is a perceived inability for agile to scale to large projects. Although studies have shown that agile methods can be scaled [21] the major thought leaders on agile believe it is not optimal [22]. When adding more personnel to a project, people factors, such as amicability, talent and skills become amplified. The average project has less than ten members, well within the reach of the most basic agile processes. Nevertheless, it is interesting to occasionally find successful agile projects with 120 or even 250 people [23]. Leading researchers show that to scale agile techniques, rules of agile have to be broken and some of the benefits are mitigated [21]. Scaling agile can have other unintended consequences that are common is large groups.

Large groups feel less ownership of project issues therefore may not engage in team problem solving. Process improvements discussed at iteration boundaries are less productive with teams of ten or more [21].

Optimum Team Size

Most of the literature indicates an optimal team size of around seven members [6][9][24]. Research suggests that large projects where members cannot provide adequate capacity need to break into separate teams to maintain optimal communication pathways [22].

Thomas Allen [9], in an empirical study of engineering organizations around the world, concluded that the probability of communication between engineers declined with increasing team size. See Figure 2. Furthermore, he stated if the team were greater than ten, the frequency showed a very modest drop in the probability of communication. Therefore, there was very little difference between ten members and fifty.

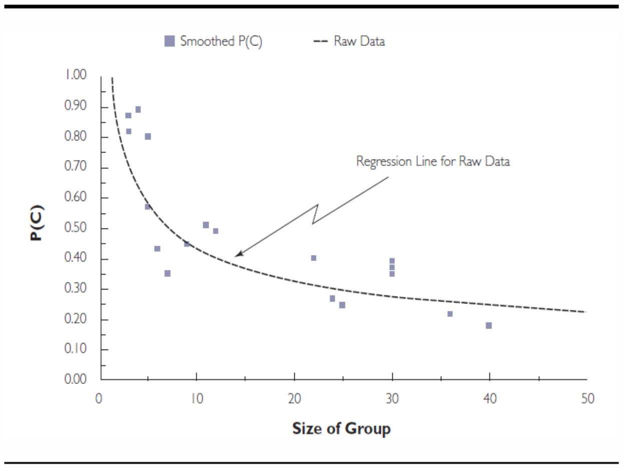


Figure 2: Probability of Weekly Technical Communication as a Function of Dept. Size [9]

As with any large project, it is also suggested to break the teams at architected boundaries. However, having teams with separate product managers runs the risk of divergent decision making. In Table 3, the advantages and disadvantages are described.

Team Size	Advantages	Disadvantages
Small	• Higher participation • Collocated teams • More likely to learn the work roles and expertise of other members • Intimate communication • Better rapport	• Less number of participants • Expertise availability • Cannot handle large projects

	• Higher cohesion	
Large	• Diverse expertise • Skills • Problem-solving approaches • Adopting different roles	• More coordination costs than smaller groups • Combining work • Arranging schedules • More dominated by the team leader • Remembering each member's particular expertise • Lower participation in group activities

Table 3: Team Size Advantages and Disadvantages Matrix [20]

Team Distance

Geographically Dispersed Teams

When searching for a solution, diverse talent pools can generate more ideas than insular ones. Organizations search for dispersed knowledge and skills to stay competitive [2][3].

A virtual team — also known as a geographically dispersed team (GDT) — is a group of individuals who work across time, space, and organizational boundaries with links strengthened by webs of communication technology [2].

There are three dimensions that characterize degrees of 'virtuality':

- Members in virtual teams are physically dispersed
- Computer driven communication is used
- Interaction is required among team members

Table 4 shows the attributes of fully traditional and virtual teams.

Fully Traditional Team	Full Virtual Teams
• **Team members all collocated**	• **Team members in different locations**
• **Team members communicate face-to-face (i.e., synchronous and personal).**	• **Team members communicate through asynchronous and apersonal means.**
• **Team members coordinate the team task together, in mutual adjustment.**	• **The team task is so highly structured that coordination by team members is rarely necessary.**

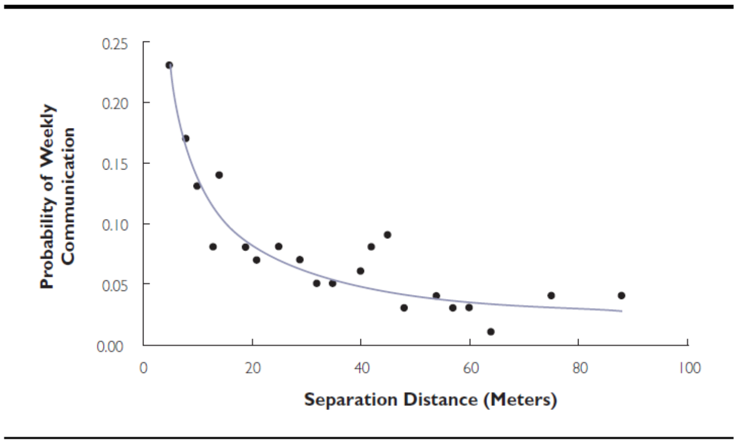
Table 4 Fully Traditional & Full Virtual Teams [2]

The side affect of having personnel in dispersed locations is the decreased richness of communication. At times, the expense of coordination becomes an offsetting factor over cost savings or diverse viewpoints, which must be considered before moving to a virtual environment [25].

Communicating using the current state of the art technology will increase the probability of communication but not replace the high-bandwidth face-to-face channels.

Probability of Communication

Thomas Allen [9], in an empirical study of engineering organizations around the world, concluded that the probability of communication between engineers declined with increasing distance. See Figure 3. Furthermore, he stated if engineers were 50 or more meters apart, the frequency showed a very modest drop in the probability of communication. Therefore, there was very little difference between 50 meters and hundreds of thousands of kilometers.



Note: Data is from seven laboratories.

Figure 3 Probability of Technical Communication as a Function of Distance Between Work [9]

However, as technology advances, communication mediums are becoming increasingly more technical [26] thereby altering Allen's curves with an increase in probability as with distance shown in figure *4*.

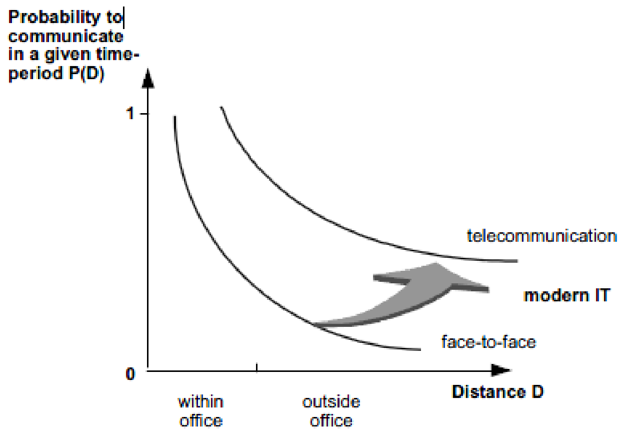


Figure 4: Probability of Technical Communication with different communication quality [26]

Affect of Team Distance

As the agile manifesto states: *"The most efficient and effective method of conveying information to and within a development team is face-to-face conversation."*

Coordinating projects over distance means an increase in overhead in order to absorb the frequent changes in requirements common in most projects [27].

Informal communication and personal relationships are important to feeling part of a team. Team distance creates an outsider psychological phenomenon of a lack of ownership [28].

Face-to-face communication is preferred over available technological solutions; therefore, agile methods are generally difficult to apply to geographically dispersed projects [29]. A leading agile consulting and training company, Construx [30], illustrates the concept with a diagram as seen in figure 5 below.

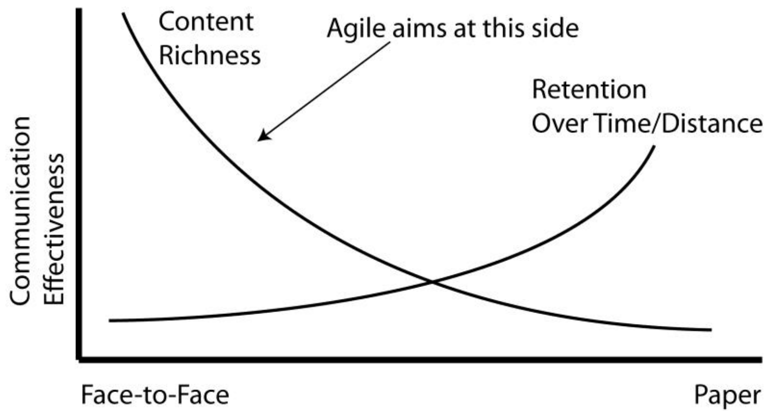


Figure 5: Communication Richness [30]

The literature shows that geographically dispersed teams have communication constraints which are contrary to agile practices. Allen's [9] study show's the probably of communication goes down with increased distance. This supports our hypothesis 2:

- *In any project, communication bandwidth is lessened with virtual teams; however, strict agile methods are impacted more negatively than strict stage gate methods.*

Industrial Trends

There has been growth in organizations converting to agile processes to manage their software development projects. Even with the backdrop of unsure executives, middle management has led the growth of agile practices and their hiring practices reflect this reality [31]. For example, figure 6 illustrates that stage gate (SDLC) type job skills (i.e. ISO9000 standards or the RUP process) have leveled off while positions requiring agile skill are still increasing.

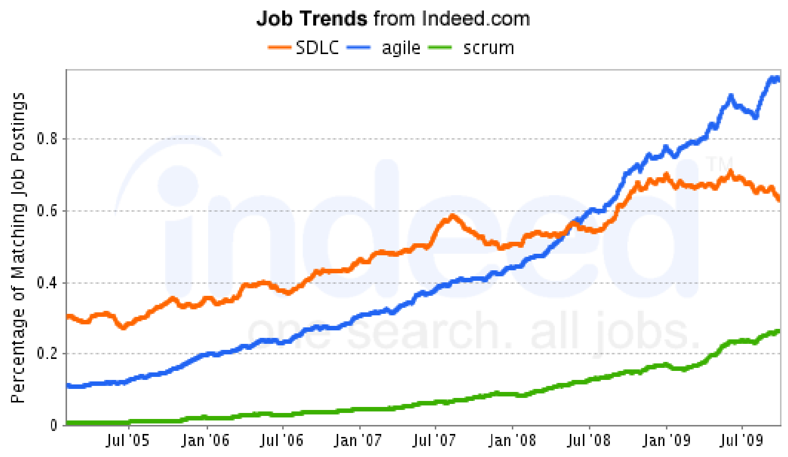


Figure 6: Agile Employment Trends [31]

The Indeed.com study data supports our hypothesis that the software industry is trending positively toward adopting agile processes.

Conclusion

Summary

The paper demonstrates the trend taking place in the software industry of organizations adopting agile processes while compensating for virtualized teams. Because agile, based on its principles, declares face-to-face as essential it is of little wonder articles and tools are emerging to address the issues.

The literature posited clearly states that smaller team sizes (roughly 7) are preferable to larger team sizes when using agile methods.

Agile methods are affective when applied to the appropriate problem set such as fluid requirements, a pervasive issue in software development. In order to deal with the fluidity, agile relies on collocation due to its communication richness characteristics.

At the Agile 2008 Conference in Toronto, Canada, one of the authors attended a session on "Virtual Agile." The speaker surprised the audience with his first statement:

"So you want to go agile and virtual, my advice is, don't do it."

He went on to discuss the struggles and mitigating techniques in order to shoehorn the two together, but the primary advice was striking.

In an extension to Allen study, Boutellier *et al* [26] showed some promise of bridging the communication gap with promising technologies, however, the gap is vast and face-to-face is still preferred.

Future Research

In our research, some books discussed the divide and conquer techniques that use architectural boundaries and multiple agile teams to mitigate geographically dispersed project problems. The technique of dividing components with clear interfaces is not new, but using it with agile is novel and may have unintended consequences such as decision divergence.

A study on how teams manage multiple agile teams on a single code base might elicit deeper mitigation techniques. An additional study using cutting edge technologies such as ‘telepresence’ systems may show the probability of communication can increase dramatically thereby making virtual agile, agile.

References

[1] W. Davidow, "*Marketing High Technology: An Insider's View*," Free Press, 1986.

[2] J. Kratzer, R. Leenders, and J. Van Engelen, "*Keeping virtual R&D teams creative*," Research Technology Management, vol 48, Mar. 2005, pp. 13-16.

[3] S. Machlis, "*Computers bring global design teams closer*," Design News, vol. 49, 1993, pp. 27-28.

[4] H. Kerzner, "*Project Management: A Systems Approach to Planning, Scheduling, and Controlling*", 6th ed., p.74.

[5] J. Phillips, "*PMP: Project Management Study Guide*," p.49.

[6] B. Boehm, R. Turner, "*Balancing Agility and Discipline*," Addison Wesley, 2003, p.167, p.28.

[7] D. Anderson, "*Agile Management for Software Engineering*," Prentice Hall, 2004.

[8] M. Cohn, "*Agile Estimating and Planning*," Pierson Education, Inc., 2006, p.28.

[9] T.J. Allen, "*Architecture and Communication among Product Development Engineers*," California Management Review, vol. 49, Winter 2007.

[10] "*Why Some Teams Succeed (and So Many Don't).*," Harvard Management Update, vol. 11, Oct. 2006, p.3-4.

[11] K. Shirahada, and K. Niwa, "*Future-Orientated Mindset's Contribute to Management of Corporate R&D Personnel Motivation in Japan*," International Journal of Innovation and Technology Management, Vol. 4, No. 4, 2007.

[12] T. DeMarco, T. Lister, "*Peopleware: Productive Projects and Teams*," Dorset House Publishing, 2nd Ed, 1999, p. 132-135.

[13] Prasad Balkundi and David A. Harrison "Ties, Leaders, And Time In Teams: Strong Inference About Network Structure's Effects On Team Viability And Performance", p.49-68

[14] J. Katzenbach, D. Smith, "The *Wisdom of Teams: Creating the High-performance Organization."* Boston: Harvard Business School, 2006.

[15] J. Meredith, S. Mantel, *"Project Management - A Managerial Approach,"* 7th ed., 2009, Dr. Anderson's ETM-545 class slides, p.15.

[16] "*3rd Annual Survey: 2008 The State of Agile Development,*" http://www.versionone.com, conducted June-July 2008.

[17] P. Abrahamsson, O. Salo and J. Ronkainen "Agile software development methods- Review and analysis". VTT Electronics 478, ESPOO 2002, p.95-96.

[18] M. Vax, S. Michaud, "*Distributed Agile: Growing a Practice Together*," Agile 2008 Conference, p.310.

[19] L. Dubé , V.l Roy and C. Bernier "*An Agile Method, a Contractual Relationship and Distance: An Unlikely Recipe for System Development Success*" Cahier du GReSI no 08-01 Juillet 2008, p.3-7.

[20] E. Bradner, G. Mark and T. D. Hertel, "*Effects of Team Size on Participation, Awareness, and Technology Choice in Geographically Distributed Teams.,"* Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), p.1-10.

[21] W. Crebs, P. Kroll, E. Richard, "*Un-assessments — reflections by the team, for the team*,“ Agile 2008 Conference, p.384-389.

[22] K. Schwaber, "*Agile Project Management with Scrum*,“ Microsoft Press, 2003, p.119-127.

[23] A. Cockburn and J. Highsmith "*Agile Software Development: The People Factor*, November 2001 (vol. 34 no. 11) pp. 131-133

[24] J. Coplien, N. Harrison, "*Organizational Patterns of Agile Software Development*," Pearson Prentice Hall, 2005, p.105.

[25] S. Horwitz and I. Horwitz, "*The Effects of Team Diversity on Team Outcomes: A Meta-Analytic Review of Team Demography*," Journal of Management, vol. 33, p: 987, 2007.

[26] R. Boutellier, O. Gassmann, H. Macho, and M. Roux, "*Management of dispersed product development teams: The role of information technologies*," R&D Management, vol. 28, Jan. 1998, p. 13.

[27] "*Using an Agile Software Process with Offshore Development*," Martin Fowler, http://www.martinfowler.com/articles/agileOffshore.html, Accessed December 1, 2009.

[28] T. Björnfot ,"*Agile Software Development and its Effects on Communication in Distributed Projects*", HUT / SoberIT 2003 Fall T76.651 Seminar on Distributed Product Development pp.1-10.

[29] P. Bertrand, J. F. cois Delisle and F. Khomh "*Toward a Framework for Agile methods in Offshore Outsourcing,*" Department of Informatics and Operations Research University of Montreal, Quebec, Canada pp-1-10

[30] "*Being Agile Without Being Extreme*," Construx Training Material, 2005.

[31] J. Moore, "*The ASPE Blog*," Available: http://theaspeblog.blogspot.com/2009/06/agile-industry-watch-agile-jobs.html, Accessed Dec. 1, 2009.