

Multi-Dimensional Management: Avoiding Calcification or Fragmentation on Distributed Teams

By Derek W. Wade and John Puopolo

Distributed agile can work, but it is risky. Even more than traditional agile implementations, teams tend to fall into two failure modes: "command and control" or team fragmentation. Distribution between teams or team members has two distinct dimensions: separation and project gravity. Distribution, including physical distance, is the most commonly recognized barrier to effective distributed teams. Gravity is the other, less recognized dimension of separation and is some stable center which teams can organize around. This can take the form of a stable product backlog, a clear mission, or an inspiring product owner. Failing to recognize where your project lies along the dimensions of separation will make it more prone to the two failure modes. Both failure modes are discussed in terms of the dimensions of distribution, with suggestions for remedial actions. In either case, leadership must be aware of both dimensions of a team/project's distribution, and act accordingly.

Perils of Distributed Agile

It is 8:30 on a cool spring morning and Jessica's soccer team is pushing for the goal.

The team, remembering Coach's advice to "control more territory," has distributed themselves around the field. This has enabled Jessica's midfielder to take the ball from the opposing team and pass it to Jessica, who deftly kicks it down the sideline. Suddenly, the other team's best player charges in her way. Jessica quickly passes the ball to her right forward – she'll be able to score! But the ball rolls out of bounds and comes to an uneventful stop.

Jessica's forward isn't there: she is five blocks away, on another field. Distribution has become fragmentation.

Distributed Agile is a naturally attractive idea. It would seem to be more bang for your buck; the success of Agile methods with a reduced cost. [CIO Insight](#) bears this out with reports that "cost reduction" is the number four goal for IT [1] with increases in spending on collaboration software for 2008 [2]. While companies *can* attain attractive returns on distributed Agile development, proponents need be aware of the risks before signing up.

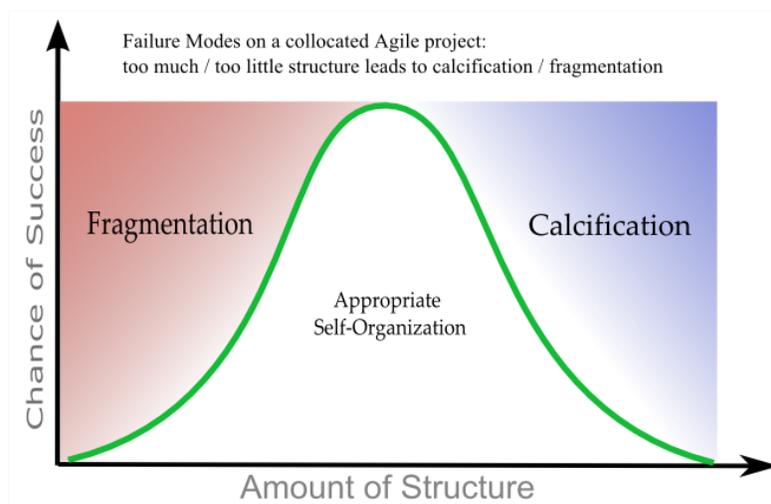


Figure 1: Collocated Agile Risks

On any Agile project, these risks manifest in two failure modes: *calcification* and *fragmentation*. Calcified teams are those that have regressed into old “waterfall” habits and have become rigid, over-structured, and process-heavy. They tend to use the administrative aspects of Agile methods, but fail to engender their rewards. Fragmented teams like Jessica and her forward, on the other hand, lack critical guiding structure. Hacking and waste are the norm; “missed passes,” poor communication, and chaos reign.

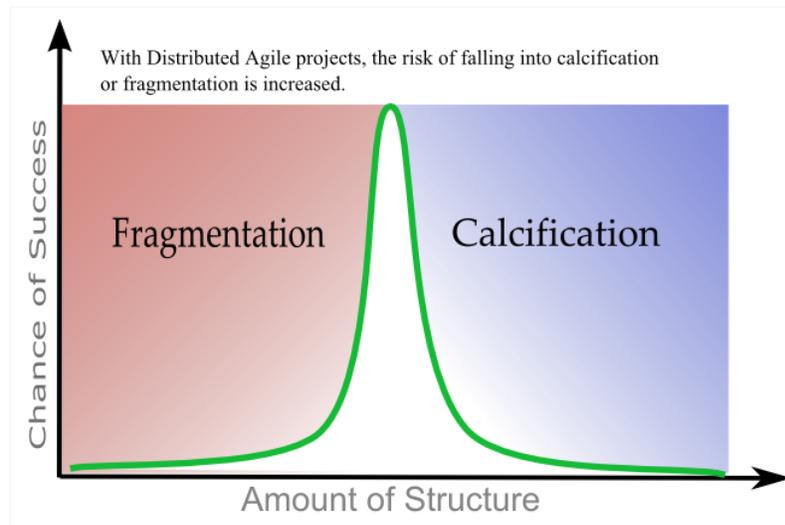


Figure 2: Distributed Agile Risks

No Agile project is immune to these failure modes, but the risk of falling into them is greater when teams are not collocated (see Figure 2.) Forces which are ordinarily negligible are readily apparent when it becomes distributed.

Root Causes – The Component Forces of Distribution

The most visible force – what most people refer to when speaking of “distributed teams” -- is the added complexity inherent in what we call *separation*. Separation comes in many forms, each having a negative impact on team dynamics:

- Geography (different cities, states or countries)
- Proximity (same building, different floors, or different buildings on the same campus)
- Time (a function of geography or work habits)
- Skill (senior vs. junior developers)
- Culture (US, UK, Italian, German, etc.)

The degree of separation extends to “lightly separated” teams who have members on-site but who are not collocated, all the way up to “hyper-separated” teams where each member is separate from the other, and balancing significant local pressures unrelated to the agile project mission. The form and degree of separation each have their own detrimental impact on the team, its ability to gel, and its overall effectiveness. In addition, it is important to recognize that their cumulative, negative effect is nonlinear due to dynamic interplay.

Despite these impacts, the globalization of companies, availability of low cost communications, and continuing use of outsourcing means that distributed teams are here to stay. High team separation is a reality of today’s business environment. We must therefore examine the other component of distribution which affects team success.

Gravity: It’s Not Just a Good Idea, It’s the Law!

A *team* is a group of individuals with a common purpose, who actively contribute and coordinate their individual talents in pursuit of a given goal. A *well-formed team* is one where the effectiveness of the team is greater than that of the sum of its members: exceptional soccer teams and jazz bands provide good examples. As practicing agilists, we have come to realize that the true power of Agile comes from the constructive bonds and interactions among members of these sorts of highly cohesive teams. This cohesion is generally most powerful when team members are collocated.

But these successful teams have one thing in common – a center around which to gravitate. This *gravity* might come in the form of a very stable backlog, e.g. when you are replacing an existing system. It might come in the form of an inspiring and effective Product Owner. Or a crystal clear mission, or a stable technology platform.

Gravity is the dimension to control when you cannot reduce separation, and there are many possible sources for this stabilizing and unifying force. You can use gravity in conjunction with the team’s degree of separation to determine your risk profile (see Figure 3).

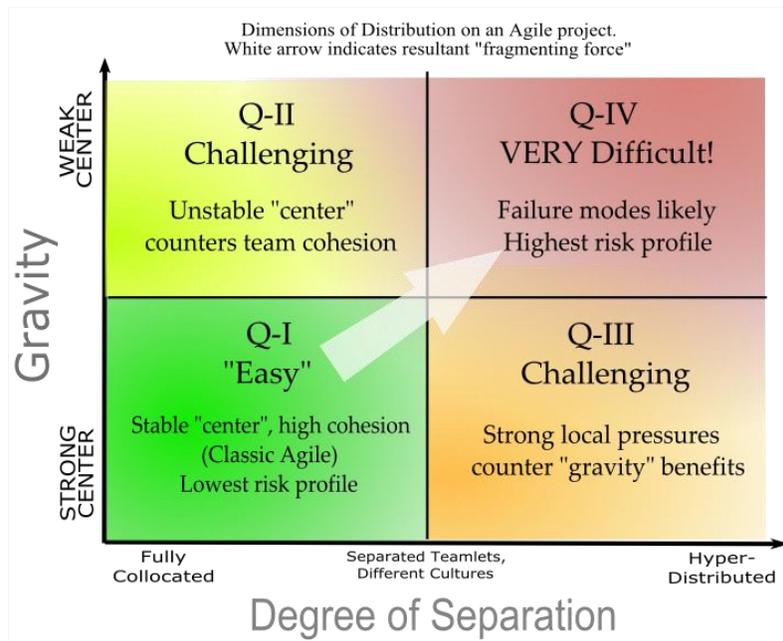


Figure 3: The Dimensions of Distribution

It is interesting to note that in the absence of a “center” with strong gravity, individual personalities magnify to fill the void. This results in the most powerful personalities dominating the team and its decisions. The watchful Scrum Master should detect this symptom and address it – and its underlying lack of gravity – immediately.

As you can see, it is best to have fully collocated teams in an environment that boasts a strong cohesive force (Q-I). As gravity decreases, and degree of separation increases, the team is at greater risk of fragmentation.

Fear of Fragmentation Leads to Command and Control

Taken together, the dimensions of distribution act as a fragmenting force on the distributed team and on the project as a whole. In discussions with distributed agile teams, we've found that project leaders who notice (or merely anticipate!) this force, unaware of its causes, tend to oppose it directly by levying heavy structure on the team.

This artificial structure subsequently leads to one of the most obvious symptoms of team calcification: *command and control*. [3] Self-organization is crushed under the "chain of command." Backlogs are broken into ever-smaller pieces, to be worked in isolation by individual team members. Simple human communication and interaction is abandoned in favor of ever more complex tools and technologies. Processes grow and mutate from agile frameworks into twisted forests of procedure and policy.

Our organic, emergent, agile team has become a collection of worker units. Our ScrumMaster and Product Owner's heads are now fully occupied with controlling the workers, with little capacity to spare for such niceties as enabling self-organization or providing a gravitational center.

When a distributed team calcifies under command and control, wouldn't it be nice to bring everyone together in a "war room" with sticky-notes and Big Visible Charts on the walls, [4] and remind them to self-organize effectively? It *would* be nice, but then the team wouldn't be distributed, and this would be a different article.

So, try the next best thing: a virtual war room. Set up a wiki page with, at a minimum: the names of team members, the current burndown graph, and the current sprint backlog items. Add any other product-related items that create or add to gravity: overall product vision, sprint goals, the Product Owner's telephone number, links to *meaningful* architecture discussions, etc.

Our virtual war room should be simple, easy to use, and easy to navigate. We have been successful using team-editable wiki sites, as illustrated in Figure 4.

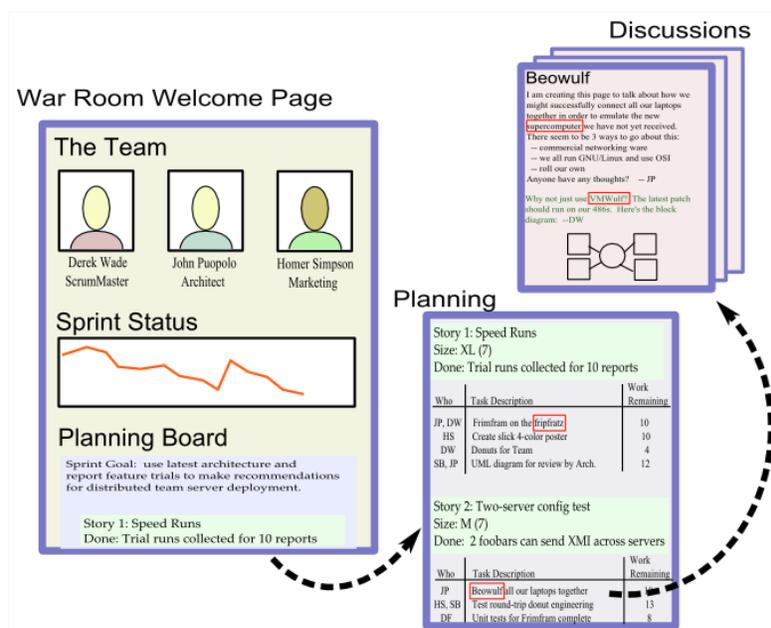


Figure 4: Using a Wiki as a Virtual War Room

In this instance, the list of team members was arranged in geographical order, from West to East. During the Daily Scrum telephone call, any team member could start, and the order would proceed “around the world.”

The other critical component of this virtual war room was a very lightweight planning and discussion system. Stories were posted on the main page of the war room, owned by the entire team. Individual team members opened new tasks by adding new lines under the story, and updated / closed a task by editing its line. Any discussion or comments about the story or tasks could take place right in the wiki.

The point here is that we’re trying to surface from command and control through lightweight, useful mechanisms that foster frequent, high-bandwidth communication. Be careful not to overemphasize tools; it just breeds more rigid structure. Focus on simplicity, meaningful communication, transparency and visibility.

The tool itself wasn’t what helped this team so much. It facilitated organic human interaction, while the team was encouraged to organize around a common center of gravity: stories with shared ownership. Reduction of separation by itself is not enough. There needs to be some center for the team to gravitate around.

Ignoring the Fragmenting Force is Abdication of Responsibility

Agile is not a license for the team to do whatever it wants, nor is it a free pass to bypass development standards and constraints. Read that again.

Agile teams are responsible for developing viable solutions within the context of the company’s values and objectives, both business and technical. In some cases, managers take the idea of self organization to an unhealthy extreme, allowing the team to make all decisions, even when they conflict with the company’s other practices.

The regrettable situation in which we sometimes find ourselves, particularly line managers and ScrumMasters, is that we listen to a voice inside our heads that goes something like this, “I am doing my specific job. The team is proactive and committed. So long as I continue to do my focused part of the Agile process, the team will self-organize and all will be well.” Lather. Rinse. Repeat.

This thinking can be taken to an extreme, where ever-so-slowly, the manager and/or Scrum Master unconsciously abdicates more and more responsibility to the comforting guise of team self-organization. The team is allowed to weaken any gravity which already exists. The management team (Scrum Master, Product Owner, business sponsor) may be hiding its head in the sand about this – albeit unwittingly – by ignoring the dimensions of distribution. The fragmenting force rules the day.

This core problem is exacerbated in distributed teams, since the social forcing functions that drive detection and subsequent adaptation are woefully underpowered. The solution is detecting reductions in gravity.

If you find yourself explaining issues and problems in terms of self organization, stop and take a close look at the team and the project, and address the real problem. Has the vision been lost? Is the Product Owner not available? Is the product backlog becoming unstable?

Management must examine the team’s ability to self organize *effectively*, its own performance and attitudes, its ability to impact positive change – and adjust accordingly. Watch out for anyone using “agile” and “self-organization” as a means of explaining away

problems and issues. This pattern of behavior is a strong smell whose root cause needs management's pressing attention. Make sure to tackle this problem at your next Daily Scrum or, at the latest, your next Sprint Retrospective. Make people aware of the situation, and work actively to apply common sense to a workable solution.

Due to the increased pressures associated with high separation, each member of the distributed agile ecosystem needs to be remarkably talented. Product Owners must be particularly dedicated. ScrumMasters must be hyper vigilant with respect to the velocity, morale, and social interactions of the team. Each team member must be proactive, communicative, emotionally mature, committed to the goal, and willing to hold one another accountable.

Manage to the Dimensions of Distribution

As leader of a distributed agile team, you are not merely trying to avoid the two failure modes of calcification and fragmentation. If you oppose the fragmenting force directly by adding more structure and control, your team will calcify under command and control. If you flee from calcification by abdicating leadership responsibility, your team will fragment while congratulating themselves on their self-organization.

You must continually stay aware of the two components of distribution – degree of separation and gravity – and work to keep your team out of the Q-IV danger area.

To help a team suffering from high separation, provide some philosophical, social and/or goal-oriented center, i.e., a team gravity, that serves as a focal point. As discussed, this gravity can manifest in many forms; but provide *something* that serves to cohere the team in the absence of physicality and strong social bonds. This gravity probably already exists in your organization and you need only make it visible; your project was started for *some* reason, after all.

If there is truly very little gravity on your distributed project, consider reducing the separation of the team until some center *can* be found and the team begins to draw naturally toward it. If you can't bring the team together physically, work to bring them together virtually.

About the Authors

[John Puopolo](#) is the founder and president of agilemania, a Boston-based agile consulting and coaching company. He'd love to hear from you at john@agilemania.com

[Derek W. Wade](#) is a Partner and Senior Consultant at [3Back LLC](#), a CSM/CSP (Certified ScrumMaster / Certified Scrum Practitioner), and Agile team coach. He has over 13 years experience across all roles of the software lifecycle in several industries including Aerospace, Distribution, Internet Commerce, Digital Mapping, and most recently Loan Financing and Employment Testing. One of Derek's distinctions is bridging the gaps between developers and business users, and his current focus is the "hard problems" of the psycho-social aspects of team product development in corporate cultures. He has led discussions on software testing and transitioning to Agile, and is currently focused on tools and behaviors which improve agile development for non-collocated teams.

References:

[1] Alter, Allan. "CIOs Rank Their Top Priorities for 2008."
<http://www.ciainsight.com/c/a/Research/CIOs-Rank-Their-Top-Priorities-for-2008>.
Accessed February 5, 2008.

[2] CIO Insight Magazine.
http://www.ciainsight.com/images/stories/Research/ItSpending/IT_2_0WhereTheMoney.jpg.
Accessed February 5, 2008.

[3] Spolsky, Joel. "The Command and Control Management Method."
<http://www.joelonsoftware.com/items/2006/08/08.html> .
Accessed April 9, 2008.

[4] The Portland Pattern Repository. "Information Radiator."
<http://c2.com/cgi/wiki?InformationRadiator>.
Accessed April 15, 2008