

2016 Project Management Symposium

Near-Site Agile: Scaling Agile with Real-World Constraints

John C. E. Johnson, PMP

Co-Founder, One Year LLC, Annandale, VA;

Masters of Engineering Graduate, Systems Engineering, A. James Clark School of Engineering, University of Maryland, College Park, MD

ABSTRACT

This paper provides a description of Near Site Agile and its advantages for scaling custom software development, especially for public sector clients. Near Site Agile splits agile teams so project managers and business analysts work onsite with the customer, and developers and testers work from nearby delivery centers where labor rates are significantly lower. This allows the developers to travel to client site as needed for agile ceremonies and client relationship building, while also providing the scalability of an outsourced delivery center for team cost, flexibility and cross-training. The trick is staying “agile” despite the lack of colocation and resulting pressures for both clients and contractors to matrix or departmentalize project teams. Lessons learned will be drawn and evaluated from IBM’s project of the year winner for project management, Electronic Records Administration (ERA) 2. This project used Near-Site Agile to deliver cutting edge custom cloud-based applications to process and access petabytes of data for the National Archives (NARA) despite significant budget, facility, and schedule constraints.

ABOUT THE AUTHOR

Mr. Johnson was one of the two Agile Project Managers on the NARA ERA 2 case study project, while he was working as a Senior Agile Project Manager at IBM. During the ten month duration covered in this project, he led the scoping and contract negotiations during the transition to Near-Site Agile. He also managed multiple agile project teams on ERA 2 using Near-Site Agile to deliver a cutting edge big data search application capable of supporting discovery across hundreds of petabytes of electronic records expected at NARA by 2020.

Mr. Johnson has since left IBM to co-found One Year LLC, a software product company operating in the DMV area dedicated to bringing cutting edge information systems to life sciences research. He has been an agile project manager and practitioner for over five years working as an entrepreneur, project manager at IBM, and a management consultant with Booz Allen Hamilton, and has led multiple award winning projects for the Marine Corps, Air Force, Navy, and National Archives.

CASE STUDY OVERVIEW

Project Title: National Archives (NARA), Electronic Records Administration (ERA) 2, Digital Processing Environment and Digital Object Repository Pilot Phase II

Short Description: The purpose of the NARA ERA 2 project was to continue development of the Digital Processing Environment (DPE) and Digital Object Repository (DOR) applications for NARA's ERA 2 program. These applications are web-based cloud applications that provide scalable transfer processing of electronic records received from Federal Agencies, as well as permanent, robust preservation and access to government records. The applications are hosted in the Amazon Web Services (AWS) cloud environment and leverage the latest in free and open source software (FOSS) components. The application has approximately five hundred to one thousand potential end users once the system moves from pilot to production (as part of a future contract).

The project started using IBM's Agile With Discipline (AWD), but was then reorganized to use a new hybrid agile approach called "Near-Site Agile." IBM's Near-Site Agile approach differs from "scrum-agile" by incorporating more formal and continuous planning. It starts with a short planning period during the beginning of the project that is dedicated to architecture and initial business requirements before development begins. Teams are made up of five developers, one business analyst, one tester, and one agile project manager (half-time), as well as one client tester and product owner. This "whole team" approach ensures that all parties involved understand the requirements and testing required for the application, and the team has the authority (through the product owner) to make decisions and self-direct changes in scope for development. Teams work within four-week long sprints where one day is dedicated to planning and one day is dedicated to a team retrospective and demonstration to stakeholders. The other 18 working days (or less) are available to develop requirements, code, and test concurrently. All work is managed as stories which are tested fully in an integrated testing environment across all previous work by application teams before the story is considered "done."

Duration: 10 Months

Cost: \$15M

Project Personnel: 35 IBM Employees, 15 NARA Employees

NARA Personnel Breakdown: 3 NARA Testers, 3 NARA Product Owners, 2 NARA PMs, 2 NARA Security Officers, 2 NARA Architects, 1 NARA Program Manager, 1 NARA COTR, 1 NARA CTO

IBM Personnel Breakdown: 15 Developers, 4 Testers, 3 Business Analysts, 3 Architects, 2 System Administrators, 2 DevOps, 2 PMO staff, 2 Agile PMs, 1 Delivery Manager, 1 Program Manager

Outcomes:

- Scoped, designed, developed, and tested a cutting-edge cloud-based archiving system in 10 months
- Delivered all "must-have" functionality on-time, under budget, and in compliance with NARA standards
- Established IBM Agile Customer Delivery Center in Rocket Center, WV
- Reduced software development labor costs by over 40%
- Winner of *Project of the Year in Project Management* in 2015 global competition

HISTORY OF NEAR-SITE AGILE

Formalizing Agile Development

The origins of Agile are based in management principles popularized during the 1980s. These techniques that would come to be called “Lean” were a combination of principles including Edward Deming's Total Quality Management (TQM), Goldratt's Theory of Constraints (TOC), and Taiichi Ohno's Toyota Production System (TPS). Lean focuses on the reduction of waste in management and production processes, by limiting the seven sources of waste (transport, inventory, motion, waiting, overproduction, overprocessing, and defects), and performing continuous improvement [5]. These complementary approaches along with the development of Rapid Application Development (RAD) by IBM in the 1980s offered alternatives to the traditional “Waterfall” management model [4]. By the mid-1990s agile development methods were beginning to manifest in the form of Dynamic Systems Development method (DSDM) which offered more structure to the RAD process that had taken the IT Industry by storm [3].

The codified agile principles were established in 2001 with the Agile Manifesto. These core principles stated that agile practitioners should emphasize the following items on the left over the items on the right [2]:

- | | | |
|---------------------------------|------|-----------------------------|
| 1) Individuals and Interactions | OVER | Processes and Tools |
| 2) Working Software | OVER | Comprehensive Documentation |
| 3) Customer Collaboration | OVER | Contract Negotiation |
| 4) Responding to Change | OVER | Following a Plan |

The first and most popular approach that fully embraced these principles was “Scrum-Agile” or “scrum.” Scrum emphasizes using small teams to accomplish small batches of work beginning at project start. This builds on the lean principles that practice proves large batches result in waste from inventory and low feedback on errors in the product process, and therefore small batches are in fact more efficient (even though this is counter-intuitive). In order to enable small batches, agile uses the DSDM principles of leveraging cross-functional teams to iteratively build software by designing, building, and testing small batches of working code using timeboxes [3].

Agile teams are therefore usually made up of a customer representative or “product owner,” a project manager, a business analyst, technical team members, and testers. To decrease process in a highly dynamic environment, all interactions should be face-to-face to quickly provide feedback and the team should “own” their work with the ability to “stop production” if an error is found and quickly fix it. Standup meetings occur every morning at the beginning of the day with the entire team so everyone knows the status and what will be done that day. This means that the purist form of agile techniques did not support off-site or distributed team approaches, because the distance would not allow for responsiveness needed to correct errors in requirements or execution for agile to be effective and efficient. However, IBM pioneered new agile approaches that attempt to get a better blend of responsiveness, predictability, and scalability.

Developing Agile With Discipline

As the largest IT services and products employer in the world, IBM needed to scale its agile processes while continuing to innovate the company delivery models. The first incarnation was the Disciplined Agile Delivery (DAD) model that advocated for the use of initial planning phases that structure the architecture and product

roadmap. With the architecture and product roadmap the technical and value propositions are known well enough to structure release points and cost estimates. This enables the agile teams to then move confidently and work in small batches that validate planning assumptions and refine requirements through the use of scrum agile techniques and structured reporting. DAD provided the ability to scale across multiple teams by having better planning processes and a shared high-level architecture [1].

The second version of this model was developed specifically for the government, called Agile With Discipline (AWD). AWD emphasized the inclusion of clients as members of the team like traditional agile, but also expanded starting and ending ceremonies for iterations (i.e. “sprints” or timeboxes) so larger stakeholder groups and SMEs were systematically included. This approach worked very well for incorporating highly departmentalized customers into a projectized delivery approach like agile.

However, AWD began to run into issues. Customer product owners worked off client-site to be colocated with the agile teams. This kept the product owners away from co-workers which led to long-time product owners losing touch with the business they represented. Also, the government owners often didn't have space to house the large number of agile teams on these scaled up projects, so separate facilities were procured for the projects. This meant that government contracts needed to be large from the beginning in order to justify new facilities dedicated to agile development. NARA was scoped tightly, had issues with available client resources, and was a magnitude smaller than previous government projects that came before it using AWD, so a new approach was needed.

THE NEED FOR NEAR-SITE AGILE

When starting the NARA ERA 2 project the original intention and contract called for using IBM's AWD delivery approach. However, challenges emerged at the start that demanded changes to the plan:

- **Lack of Product Owners** – NARA had been under a hiring freeze and didn't have enough experienced and senior staff who could be spared to be full-time product owners for ERA 2
- **Lack of Space at Client Site** – NARA originally planned to have space for IBM's development teams, but AWD requires open “agile” spaces and dedicated conference rooms to enable face-to-face interaction which were not available due to a lack of space sharing and regulations on work station modifications
- **Lack of Talent Near Client Site** – IBM identified that NARA's location in College Park was not easily accessible by its available developers; most of whom didn't have the skills needed for the ERA 2 project

These challenges were critical. Each challenge hobbled the ability of the team to commit to and take advantage of the core principles of agile development: self-direction, colocation, and whole teams.

A lack of product owners meant the project would either lack proper direction by the client or the ability for teams to self-direct. Teams would either have a product owner with no leverage or insight, or the team would be severely limited so NARA could “approve” and “review.” NARA's lack of available resources also impacted the ability to have NARA testers on each of the agile teams to enable continuous delivery of “accepted” code.

The lack of space meant that teams would not be able to communicate quickly and effectively through open interaction and face-to-face communication. This would significantly slow down the teams and limit their ability to iterate designs quickly and resolve potentially competing designs. Without colocation the teams would be highly limited to respond to change through fast interaction with each other.

The lack of talent near NARA threatened the ability to colocate as well as put together whole teams of people with the required skills. IBM had recruited mostly near Herndon, VA, a two hour drive in DC rush hour traffic to NARA's College Park site. At the same time, IBM had been cutting costs through very high utilization targets across all departments, especially those in software development. This meant that many projects needed good developers and resultantly the best talent could choose to work at projects close to home. The lack of available talent was potentially crippling for IBM, since ERA 2 was scoped to include the state-of-the-art cloud-based application development. All members of the ERA 2 had to be strong programmers capable of learning new technology quickly, so either IBM was going to have to pay high prices subcontractors (and take on the associated risks) or pay for developers to stay in nearby hotels to reduce their travel burden.

Faced with nearly insurmountable odds against using agile development but requiring agile per the contract, IBM worked to devise a new strategy using what it knows best – information technology and daring project management techniques.

IMPLEMENTING NEAR-SITE AGILE

It quickly became clear that there was no solution for many of the geographic and facility space constraints at the NARA client site. At the time this became clear, there were many pressures to modify the contract and go back to a waterfall-style plan. Waterfall would enable the team to work remotely without great penalty and deliver within budget. This however could not work because NARA lacked design specifications, and there wasn't enough time in the schedule to go through NARA's governance for traditional IT projects. IBM needed to come up with a solution that could still deliver on-time and in-budget using iterative development.

The IBM project executive sponsor proposed a new concept, “Near-Site Agile” as a solution. This approach would leverage a nearby client delivery center (CDC) in Rocket Center, WV that was performing O&M support for NARA already and outsource the development. Rocket Center is a small town, but there was enough talent immediately available at the CDC since IBM was the premier and only large IT employer with access to UMD's Frostburg campus. The management team worked together with NARA and the CDC to come up with a solution that could work. The principles were as follows:

- **Colocate by Function** – Teams would split geographically between business and technical roles
 - Project managers and business analysts would locate at client-site
 - Developers and testers would locate at the client delivery center (CDC) in Rocket Center, WV
 - Teams were kept “whole” and would still maintain the power to self-direct
- **Near-Site Team Building** – teams would travel to each other's site when needed for team building
 - Teams would start off colocated to build strong relationships before splitting by function
 - Teams would travel to colocate monthly for iteration planning, midpoint checks, and retrospectives
- **Simulated Colocation** – teams heavily use video-conferencing for meetings and quick conversations
 - All team members would be equipped with high-definition cameras at workstations
 - NARA conference rooms were outfitted with HD cameras and audio equipment

“Colocation by Function” enables teams to have the delivery center benefits of training across projects and sharing of ideas between those team members that do similar types of work. Because the customer is the team's source of requirements, the requirements leads (project manager and business analyst) must co-locate at the customer location. This enables greater responsiveness and coordination between the customer and off-site technical teams. *It's very important to emphasize that collocating by function does not take away the*

team ownership of the code, especially by the developers. Each team member remained empowered and able to self-direct their work despite being dispersed from the client.

The “Near-Site Team Building” requirement ensures that when needed all team members can be co-located for workshops, planning sessions, and review sessions. This helps build and maintain the trust that is required with agile delivery and the speed of communication needed for highly feedback intensive meetings.

“Simulated Colocation” makes sure that teams can move fast. Stand ups, requirements meetings, and even simple discussions for feedback on testing or design options are close to in-person quality with HD video-conferencing. Software like Google Hangouts that have one-click conferencing abilities lowers the inertia so no conversation requires more than a single action to initiate face-to-face communication.

LESSONS LEARNED

Challenges with Near-Site Agile

While there are many benefits to Near-Site Agile, there are some cautions a project manager and management team needs to be aware of before employing the approach. These focus on the inherent multi-site and off-site challenges that must be addressed directly, as well as the balance of investment in the near-site client delivery center (CDC):

- **Travel and Communication Costs** - Logistical challenges with this approach are primarily rooted in maintaining cost efficiency of distributed teams despite travel on a monthly or more frequent basis by all team members. The Near-Site Agile approach also requires significant setup of collaborative technologies in both locations, which can add cost to the customer. Finally, the ability to distribute teams is limited to a reasonable travel distance, so that teams still meet within the same or nearly same time zone. This enables coordinated standup meetings in the mornings and self-forming teams as needed to support feedback loops central to agile project processes.
- **CDC Incentives and Culture** - From a teaming perspective, the travel does help to normalize cultures across locations, but only it's not a perfect remedy. Part of the benefit of a client delivery center is a low-cost labor market which is often centered around a community where the community members have deep roots. This adds significantly to the loyalty and stability of team members on the project, but adds the intricacies of small town politics. The biggest challenge is accurate and timely escalation of issues and making sure that team members are not overworking themselves to compensate for bad planning in order to bolster the CDC reputation. The PM must be actively fighting these tendencies that could lead to team burnout and inherently exploit the CDC teams.
- **Talent and Growth Rates at CDC** - Lastly, the benefits of using Near-Site Agile can be great, but there is a limit to the growth rate sustainable at a CDC in terms of available talented resources. While at first there may be large untapped talent in a small town near a major metropolitan area, eventually the growth rate of the projects at the CDC becomes greater than what the small town can support. This happened within the ten months of the NARA ERA 2 project. To address this issue, CDC members need to train each other to ramp up skills acquisitions and the PM needs to be prepared to send senior staff to to the CDC to train and grow teams.

Benefits of Near-Site Agile

The first and primary benefits of Near-Site Agile lie in its low-cost, stable resources. The incredible support that was received from the Rocket Center client delivery center (CDC) was at the core of IBM's success on the NARA ERA 2 project. However, there are also inherent advantages that a Near-Site Agile delivery offers beyond standard on-site disciplined agile models by reducing client burdens. These benefits more than make up for the challenges in employing Near-Site Agile:

- **Low-Cost Resources** – savings on software development labor can be 40-50% or more. This more than makes up for the additional logistical costs and can be an incredible competitive advantage. In agile, the labor mix is heavily technical. With five out of six and a half team members from the CDC (each time has half a PM) the net labor savings can be up to 40% compared to staffing at client-site.
- **Reliable Resources** – The CDC in Near-Site Agile is an incredible opportunity for the town where it's located. People at the CDC stay in the town because that's where they want to live. This means that resources are stable and the PM spends less time in-project training or backfilling.
- **Stable Product Owners** – client personnel, especially product owners, can now remain on-site and still be completely connected to team members using video-conferencing. This allows product owners to stay engaged at least part-time in the business they represent – increasing their effectiveness.
- **Reduced Facility Costs** -because the CDC is located in a small town with a lower cost of living, the facilities are cheaper and often flexible to be designed as true “agile spaces.” The benefits of open designs and dedicated team rooms become quickly apparent as teams spend longer getting things done, rather than finding each other or bouncing between conference rooms.
- **Scalability of Delivery** – because the Near-Site Agile approach lowers the burden on the client for facilities and people there is more actual room for scaling up delivery, especially in government environments. With lower burdens on product owners more client personnel will be willing or even want to become product owners once they see success in the first releases. And the CDC can grow through resource sharing to meet that demand efficiently until human resources becomes a constraint.

REFERENCES

1. "Disciplined Agile Delivery : A Practitioner's Guide to Agile Software Delivery in the Enterprise." *Disciplined Agile Delivery : A Practitioner's G...* N.p., n.d. Web. 14 Mar. 2016.
2. "Manifesto for Agile Software Development." *Manifesto for Agile Software Development*. N.p., n.d. Web. 15 Mar. 2016.
3. "A Quality Software Process for Rapid Application Development." *ResearchGate*. N.p., n.d. Web. 15 Mar. 2016.
4. "What Is Rapid Application Development? (RAD)." *What Is Rapid Application Development? (RAD)*. N.p., n.d. Web. 15 Mar. 2016.
5. Womack, James P., and Daniel T. Jones. *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. New York: Free, 2003. Print.