Information Management for National Guard Agribusiness Development Teams: An Agile Development Case Study

Authors: Lynndee Kemmet, Network Science Center at West Point; Ray McGowan, Army CERDEC; C. Reed Hodgin, AlphaTRAC Inc.

Abstract:
One of the grand challenges in a modern tactical environment is providing our warfighters with capability in a rapidly evolving and highly dynamic mission space. One of the military domains that has a great need but minimal capability are the soldiers who support socio-economic development lines of operation within the U.S. Department of Defense. In an attempt to meet this challenge, a team including the Kansas National Guard, CERDEC and the Network Science Center at West Point worked with DARPA to create a knowledge management system that will begin the process of acquiring the data to support systems research, systems thinking and systems engineering for socio-economic development analysis by the Army, academia and industry. By use of an agile development process, a rapid development process successfully fielded new information capabilities to National Guard Agri-business Development Teams (ADTs) within one year. This new capability in theater allows for a more systematic collection of data, which will be used to support enhanced analytics for non-kinetic lines of operation.

David Kilcullen, in *The Accidental Guerilla* said it well when he argued that we have entered a new era of warfare in which the "enemy" is often not nation-states but non-state actors who are far more agile and adaptive than large nation-states. As a result, success in this new era of warfare will require that we also become more agile so that we can adapt quickly to changing environments.

American soldiers and their commanders are not unaware of this need to adapt. They have witnessed first-hand on the ground the ability of the "enemy" in both Iraq and Afghanistan to adapt quickly to changing circumstances in order to overcome efforts to defeat them. Insurgent groups have shown themselves to be mobile, capable of using simple technology to thwart sophisticated western technology and adept at quickly using what information they gather to guide their actions. Unburdened by any rule-bound system on how they must acquire and use resources, insurgent groups are often one-step ahead of coalition forces.

As in any conflict, one of the most valuable assets is information. Military forces of most all nations, including the U.S., have been well aware of this and the collection and management of information has long been seen as critical for the successful planning and execution of missions. Advances in technology have made it possible for modern militaries to collect a mass of data but process has not kept pace with technological advancement.

Despite all the new sophisticated technology available to coalition forces in Iraq and Afghanistan, insurgents still often seem to know things sooner and act on that information faster than do coalition forces fighting them. The collection of information has become much easier, but the "actionable" use of that information remains a challenge. Soldiers and commanders are now overloaded with data and sifting through it to find what is of a value and what is not at any given moment is difficult. In addition, the sharing of information remains a challenge, partly due to technological barriers in sharing information among systems and partly due to policy barriers related to information sharing.

Civilian and military researchers are well aware of these challenges and the need to overcome them. Millions of dollars are being poured into the effort to find technical solutions to information overload and to overcome roadblocks to information sharing. But what is often overlooked is the need to overhaul the "process" by which these potential technological solutions are developed and delivered to soldiers on the ground. It is this part of the acquisitions process that needs modification.

Far too often, the process is one in which technology developers, whether civilian or military, determine what they see as the problem or need – the requirements – and then develop what they see as the solution to the problem they have identified. That solution is then "tossed over the wall" to soldiers and often the result is that soldiers toss the new technology to the side, either because the technology addresses what they feel is the wrong problem, or a non-problem, or doesn't adequately solve the problem. If developers are lucky, solders will find an alternate use for that technology, which means that not all of the time and money spent in technology development is lost.

One thing that has contributed to this mismatch between technology development and actual soldier needs is timing. The technology development process in the defense industry is so long and cumbersome that by the time a new technology reaches the field, the situation has changed and that technology is outdated and no longer addresses the new needs in the field. The technology development world is not unaware of this problem and is increasingly talking about "agile" development.

Agile development, as described in a 2001 *Manifesto for Agile Software Development*, essentially is a process that emphasizes interaction between the developer and the user throughout the development process. Developers remain open to adapting design based on user feedback. This process should be rapid, meaning that users provide feedback and designers make quick adjustments and toss the ball back to users for quick feedback. Ideally, agile development results in technologies that reach users much more quickly. And because users are part of the development process, the technology given to them should actually meet their needs. If the user's needs change during the development process, technology developers should get that new requirement from users and adjust during the development process. This should prevent the fielding of technologies that address outdated requirements.

What would such an agile development process actually look like? Such a process was recently used in the development of technologies to assist National Guard Agribusiness Development Teams (ADTs) operating in Afghanistan. ADTs, company-sized units that combine agriculture subject matter experts with the team's own security force, are one model of military efforts to conduct stability and reconstruction missions. Such missions, although essentially non-kinetic, are a key component of overall U.S. security policy. Like units on more traditional, kinetic missions, ADTs face the challenge of operating in rapidly change environments and have a need for timely and relevant information for situational awareness and mission planning.

With funding support from the Defense Advanced Research Projects Agency (DARPA) and the office of the Secretary of Defense, and under the guidance of the ADT Project at the Network Science Center at West Point and the Communications- Electronics Research Development Engineering Center (CERDEC), two technologies were developed specifically geared for us by units on ADT-type reconstruction missions. Both technologies were developed with constant interaction and feedback from ADT users. And both were fielded within the first year of development by leveraging previous technologies and rapidly responding to user needs.

National Guard Agribusiness Development Team Knowledge Management System: A Case Study of Agile Support for Non-Kinetic Effects

One of the emerging challenges from the mission command domain is the ability to leverage the data and information available across the tactical domain to support decision- making by commanders. The "data-to-decision" paradigm is challenging but effective for most traditional warfighting domains. This is so because the user community and the domain subject matter experts understand what are the critical information requirements that support their workflows and how these information requirements serve as either lagging or leading indicators towards some objective within their workflow. In some cases, there might even be knowledge about how these indicators tie to overarching warfighter outcomes within a particular mission or campaign plan. This is a fair characterization for the kinetic side of the tactical warfighting.

However, on the non-kinetic side of tactical engagements, the picture is not as clear. In most areas that fall outside the tactical intelligence domain about people and infrastructure, the US military has a small knowledge base that it can use to address the complex interactions and inter-relationships that occur between the entities in the Political, Military, Economic, Social, Infrastructure and Information (PMESII) domain. One of the reasons for this is that irregular warfare and stability operations have forced the US Army to support many missions, such as economic development, that traditionally would be performed by chartered government agencies like the Department of State or United States Agency for International Development (USAID). This is so, for example, in Afghanistan because of the instability of conditions. Although tasked with this reconstruction mission, the military has neither the information nor the tools to carry out critical tasks in a complex mission space.

How this challenge might be overcome is addressed in this case study presented here. The application of an agile systems engineering approach to the development of knowledge management systems designed for use by ADTs in Afghanistan helps highlight the attributes of an agile systems approach. This paper describes the use-case for the National Guard Knowledge Management project and how agile systems engineering provides an innovative methodology to enhance the mission effectiveness of the ADTs and capacity building, in general. Through this project, lessons have been learned and insight has been gained with regard to how best to apply an agile development approach.

Agile Systems Engineering

The goals of agile software development and agile systems engineering are complimentary with agile software development being an aspect of the larger agile systems engineering process. The agile manifesto which encompasses the guidelines used by the agile software development domain focuses on the following values:

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.

It is clear to see how these values directly apply to any effort where the primary objective is to produce rapid value for the intended customer. The customer is dedicated, knowledgeable, and represents the typical user and can prioritize high-value functionality and the architecture is designed for current requirements where the requirements for the system are largely emerging and can change rapidly. These values lead to the 12 principles that guide agile development:

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity, the art of maximizing the amount of work not done is essential.

11. The best architectures, requirements and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

In applying the agile manifesto to systems engineering one wants to 1) increase the speed that new products and systems are designed and delivered to the customer, while 2) addressing the uncertainty in future user needs, operating conditions and functional requirements. The best way to achieve this is by focusing on the flexibility and speed of the process for moving from concept to implementation of the targeted customer systems/capability. This is the major difference between keeping the customer focus by delivering software that meets the customer's need to a focus on the process that allows the developer to meet the customer goal of providing agile software in a timely manner to meet dynamically evolving customer workflow requirements.

Development of XCapture (Automated After Action Review)

XCapture is an automated After Action Review technology that is designed to improve the information collected about missions in order to improve future mission effectiveness. In essences, it seeks to:

- Capture event and experience
- Address what was supposed to happen
- Report what really happened
- Ask what did we learn
- Archive the information in a useful way
- Make creation of reports automatic
- Archive for use in Knowledge base / AI development / training / operational decision support

The first step in the development of XCapture was to leverage existing technology known as AlphaACT. AlphaACT is the outgrowth of a DARPA SBIR. It is a hosted, web-based software capability that goes beyond data management to support and improve the decision-making process itself. The system uses Artificial Intelligence to capture and use the experiences of successful decision-makers and it is designed to operate in both training and response modes for crisis management and - in development - battlefield applications. AlphaACT also creates a community-wide knowledge base, increasing the sharing of knowledge and experiences among decision makers. AlphaACT is currently operational for emergency response training and is in development for small unit leader decision training.

Why was an agile development process applied to the creation of XCapture? Because commercial software product development experience in the last 15 years has shown that an adaptive approach helps ensure both project and end user satisfaction. The agile development methodology has been shown to be most effective in this environment.

In the agile development process, “user stories” are used to establish the needs and requirements of product users and ultimately drive product development and modifications. The scrum process is used to iteratively and incrementally turn the user stories into functional capabilities.

The scrum process involves a series of cycles (called sprints) lasting two to four weeks during which a number of user stories are selected for development and test. Using the scrum process allows the development team to respond quickly to emerging and evolving product needs and to address bugs quickly and with structure.

This type of development requires independent testing, verification, and, if necessary, re-verification of each development task or item. The functionality or item is then presented to the entire team at the end of the sprint to demonstrate that it works and to gain surrogate user acceptance.

In addition to the testing done within each sprint, after each major release of product improvements, the entire team breaks from the development process to thoroughly test the system. System errors found during this phase are reported to the development team for correction and are retested before the new release is made available to beta testers. Only then does a new cycle begin.

The process applied to the creation of XCapture that resulted in development to deployment in less than one year followed this process:

- Gather user stories and let them guide initial development:
  As a user, I need the system to _____ so that I can _____

  Example user stories include:

As an end user I need the AAR tool to be targeted at the ADT team members and team leader as primary initial users so that the most important users can be served first.

As an end user I need the AAR tool to quickly capture the commander's intent and desired end state so that others and I can understand what was supposed to be accomplished on a mission. (story) (Immediate) (High priority).

As an end user I need the AAR tool to quickly capture the mission plan / order so that others and I can understand what was supposed to happen on a mission. (story) (Immediate) (High priority).

- Sprints
  - Short development cycle (2 – 4 weeks)
  - Operating product for testing at end of each sprint
- Establish end user advisors / collaborators
- Brainstorm initial needs
- Prioritize needs
- Kick off sprints
- Conduct sprint
- 2 week cycle
- Working product update
- Hold sprint review session with end users
- Demonstration
- Play with it
- Get end user feedback
- Establish new user stories
- Reprioritize
- Repeat

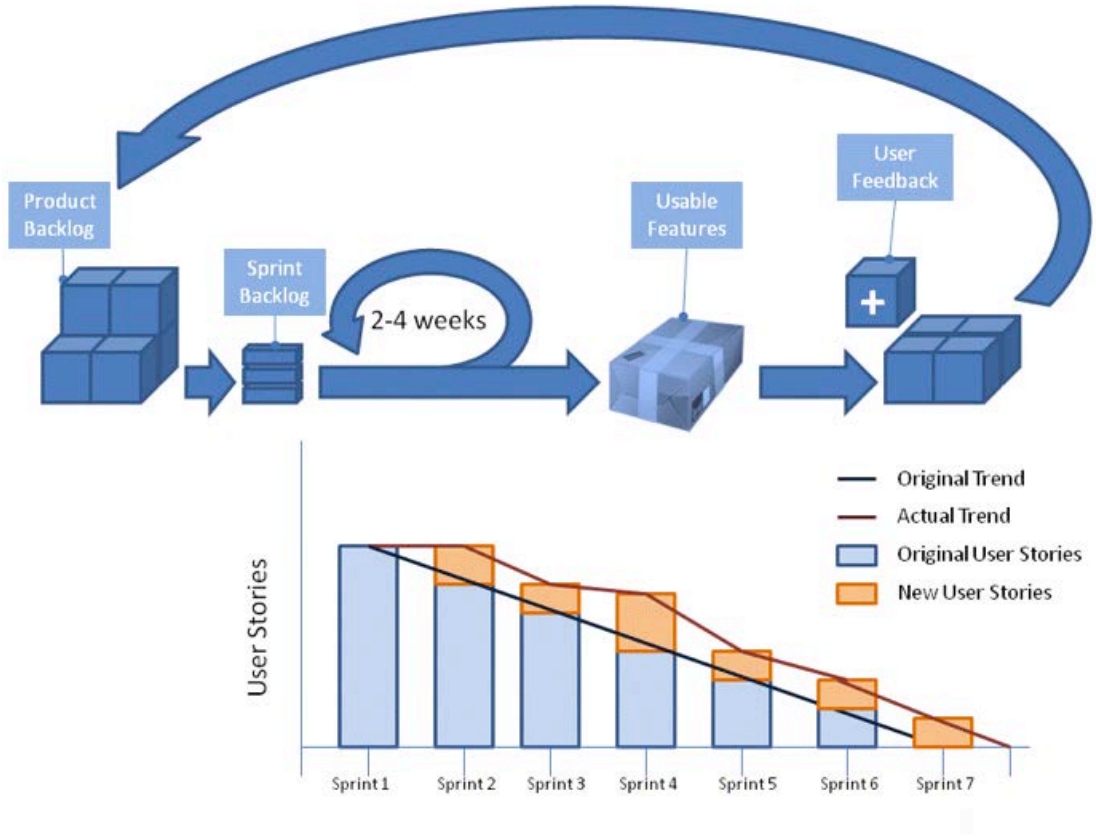

Figure 1 Agile product development process

This process described above was effectively used to develop and deploy both XCapture and project planning tool called mobile Task Assistant. Mobile TA also leveraged an existing technology platform – Task Assistant. And, again the development process started with gathering information on an unmet user need, in this case an automated process for collecting and reporting data needed to generate the documentation for the approval of project using funds through the Commander's Emergency Response Program. As with XCapture, mobile TA was developed through constant interaction with and feedback from users.

Conclusion

Traditional technology development followed a classical format that involved creating requirements – often without input from end users, building technology to meet those requirements, tossing the new technology "over the wall" to soldiers and then leaving them to "like it or lump it." The result has typically been the development of technologies that users neither liked nor needed. Or, more accurately, the development of technologies no longer needed because the situation had changed long before the technology ever made it from drawing room to the field.

The agile development process takes a different view, which essentially is that the end-user – in this case the warfighter – is all that matters. What technology developers think they want, or should want, is not what matters. This agile approach requires constant interaction with users during the development process with constant adjustments as the needs of the user change. As this case study of XCapture and mobile TA show, when effectively applied, the agile development approach can lead not only to faster deployment of new technologies, but also to the creation of technologies that better meet the needs of users and are more favorably viewed, and therefore more widely used, by warfighters.