

What Lessons Can the Agile Community Learn from A Maverick Fighter Pilot?

Steve Adolph
University of British Columbia
Vancouver, B.C.
Canada
steve@wsaconsulting.com

Abstract

For the agile software development community, agility is defined by the values expressed in the agile manifesto. But in concrete terms, what does it mean for a software project to be agile? US Air Force Colonel John Boyd defined agility as the ability to operate the Observation-Oriented-Decision-Action (OODA) loop faster than an adversary. Agility therefore depends on the tempo at which we can exploit the OODA loop, and it is culture, not methodologies or tools that determine our OODA loop speed. This definition of agility has implications for the software development community. This short paper introduces Colonel Boyd, the OODA loop, the factors which influence OODA loop speed and the possible research opportunities into software engineering culture we are considering.

1. Introduction

Agility is often defined using metaphoric terms like nimbleness, lightness, ease of movement which brings to mind the graceful and seemingly effortless movements of a gymnast or ballet dancer. In a business context, agility is usually defined in terms of timeliness and flexibility. For the agile software development community, agility is defined by the values expressed in the agile manifesto.

However, what does it really mean for a software development organization to be agile? If a project adopts an agile methodology such as XP [1] or Scrum [2] is it agile? Jim Highsmith is quoted by Pete Mcbreen as saying “it may say agile on the box, but it doesn’t feel agile” [3]. Alistair Cockburn cautions that simply using iterations, user stories and velocity doesn’t mean your project is agile - or on the way to success [4]. On the other hand if a project follows a legacy methodology such as RUP or even waterfall

based methodologies is it by definition not agile? Many RUP and UP experts claim their process is agile [5, 6, 7] and many practitioners (including this author) will enthusiastically claim that even though they were not following an agile methodology, their projects were agile. So what does it really mean for a software development organization to be agile?

US Air Force Colonel John Boyd (1927-1997) may have an answer to this question. This paper answers the question what is agility by introducing Colonel Boyd’s Observe-Orient-Decide-Act (OODA) decision model and how he defined agility in terms of his time based decision model. Next we draw on Colonel Boyd’s principles for operational success to identify the factors that influence agility within an organization and then examine their implications for software development. Finally we briefly describe the research opportunities revealed by Colonel Boyd’s work that we are exploring.

2. Colonel John Boyd

Colonel John Boyd fit the stereotype of the US fighter jock, big, loud, confident to the point of arrogance, and a superb pilot [8]. He earned the nickname “40 Second” Boyd from his standing offer to pay forty dollars to any pilot who could defeat him in a dogfight in less than 40 seconds. Legend has it that no pilot ever collected. If Colonel Boyd was just another superb fighter pilot, then he would be of little interests to agilists—at least to those who are not pilots. But Colonel Boyd was much more than a superb pilot; he was also a superb strategist.

While developing his air-to-air combat theories Boyd developed new ideas about strategy which he presented starting in 1977 in his “Patterns of Conflict” briefing [9]. In “Patterns”, Boyd generalizes his air-to-air combat theories and argued that to win in a competitive environment we must operate at a faster tempo or rhythm than our adversaries-or, has he

explained it, get inside the adversary's Observation-Orientation-Decision-Action time cycle (O-O-D-A, or OODA) [9]. Boyd defined the ability to “operate inside an opponent’s OODA loop as agility [10 p.66]. This leads to the questions, what is the OODA loop and what does it mean to operate inside an adversary’s OODA loop?

3. The Observe-Orient-Decide-Act Loop

Imagine two fighter pilots locked in a dog fight, each pilot scanning the skies, observing the other. As the pilots observe each other they draw on their training and experience to continuously construct and revise their mental models of the situation attempting to orient themselves to the unfolding events. The pilots act without thinking, decision and action flow through from their orientation. Each pilot works to gain a position of advantage using quick maneuvers attempting to disorientate the other. Suddenly, the opponent appears where he was not expected, a flash, and one pilot flies away victorious. This melodramatic scenario attempts to capture the intuitive essence of the OODA loop, Observe, Orient, Decide, and Act. Figure 1 is a graphical representation of Boyd’s OODA loop.

The OODA loop is most definitely not a linear staged decision making model and while it is colloquially referred to as a loop; it is a misnomer to think of it as a single loop. Observation, orientation, and action occur continuously and simultaneously. Orientation becomes the most important part of the OODA loop because it shapes the way we observe, the way we decide, and the way we act. Boyd described orientation as representing the “images, views, or impressions of the world, shaped by genetic heritage, cultural tradition, previous experience, and unfolding circumstances [11 p.13]. We act in accordance to how we perceive the world and not how it is.

Boyd defined agility as: one side in a conflict or in competition is more agile than its opponent if it can change the situation more quickly than the other side can update its orientation. He described this as the ability to operate inside an opponent’s OODA loop [10 p.66]. Agility therefore depends on keeping one’s orientation better matched to the real world than one’s opponent during times of ambiguity, confusion, and rapid change, when one’s natural tendency is to become disoriented [10 p.30]. While agility is about using time for competitive advantage it is not speeding up the loop by doing things faster because doing the wrong things faster is not a strategy for success.

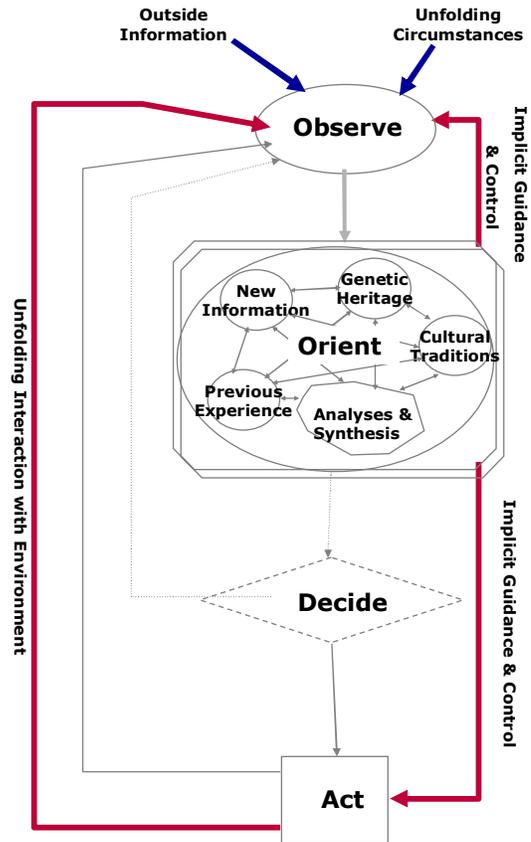


Figure1 Boyd's Observe, Orient, Decide, Act Loop [12]

If agility is our ability to get inside an opponent’s OODA loop then we must understand the forces that influence our ability to operate our OODA loop more quickly than our opponent.

4. Agility, Friction and OODA Loop Speed

On May 9th 1940 the world witnessed the vivid example of the German Blitzkrieg when Germany began its attack on France. Less than one month later, the last of the British troops were evacuated from Dunkirk and Germany had defeated France and Britain. Contrary to what some may believe, Germany did not have superior firepower or troop numbers. Without superior equipment, or superior numbers how could Germany seemingly without effort defeat France?

According to Boyd, the root of this success lay in the German system for dealing with people and therefore the strategy for winning was cultural rather than technical. This cultural advantage enabled the

Germans to execute their decision loops more quickly and more inconspicuously than the allies and create confusion and disorder in their defense forces. According to Boyd researcher Chet Richards, Boyd called these cultural talents the Organizational Climate for Operational Success”, sometimes referred to as the “Principles of the Blitzkrieg” [10 p.51]. Boyd used the German terms for these talents and referred to them as Einheit, Fingerspitzengefühl, Auftragstaktik, and Schwerpunkt: [10 p.51]. In English we could name these talents as unity or trust, skill or expertise, intent, and vision.

Unity/Trust
(Einheit) Unity or mutual trust is the internal harmony that converts a mob into a team. Trust develops from common experiences and mutual respect.

Skill/Expertise
(Fingerspitzengefühl) This is a zen-like quality of intuitive understanding that results from years of training and experience and enables an individual to “just know what to do” in changing and uncertain situations.

Intent
(Auftragstaktik) Intent or the contract of leadership: a commander gives their subordinate a clearly defined goal and obtains their agreement to accomplish the goal. The command does not prescribe how the subordinate should accomplish the goal, rather the subordinate is given freedom to take the initiative which enables flexibility and creativity in execution. In case of controversy, the superior is expected to support the subordinate, so long as the subordinate operates within the “contract.”

Vision
(Schwerpunkt) The concept that gives focus or direction to our efforts. In Boyd’s words, “Schwerpunkt represents a unifying concept that provides a way to rapidly shape focus and direction of effort as well as harmonize support activities with combat operations.

It is easy to see how these aspects mutually support each other to create a climate for organizational success. Unity and trust are the moral forces that harmonize a group by enabling people to subordinate

their personal goals to the team goals and the goals of the organization. With trust, people feel safe taking the initiative rather than worrying how to justify their actions. Skill and experience enable people to clearly observe their environment and develop creative solutions. Klein’s research shows how skill and expertise enable people to operate quickly in uncertain environments [13]. Intent opens the opportunity for people to take the initiative. Vision gives people direction, in those situations where no firm guidance has been given (which should be most of the time).

Compartmentalization, alienation, non adaptation and fixed recipes slow down the loop. The common theme in these is disconnection. Our orientation becomes disconnected from the environment and we are unable to correctly observe the environment, resulting in our orientation becoming corrupted with incorrect data. Anything which hinders Observation in any way risks not only feeding bad information into Orientation but corrupting all of the other processes in the loop [10 p.63].

Prussian strategist Karl Von Clausewitz used the term friction to describe the atmosphere of war. “.....the influence of an infinity of petty circumstances, which cannot properly be described on paper, things disappoint us, and we fall short of the mark...Friction, as we refer to it here, is what makes the seemingly easy so difficult.” [14 p.164]

Boyd used von Clausewitz’s term friction when summed to characterize the forces that influence OODA loop speed, “friction is generated and magnified by menace, ambiguity, deception, rapidity, uncertainty, and mistrust. Friction is diminished by implicit understanding, trust, cooperation, simplicity and focus... harmony and initiative tend to diminish friction” [11 p.9].

A strong cohesive culture reduces friction and enables quicker execution of our OODA loop. Quicker execution of our OODA loop enables us to operate more quickly than our adversary and we become agile. Agility is a cultural phenomenon and organizational practices based on Boyd’s principles for operational success encourage an agile culture.

5. Is the OODA Loop Suitable for Software Development?

Much of the power of the OODA loop lies in the mental and moral dimensions, the confusion, mistrust, and disorder created in an opponent when one force is able to cycle through their OODA loop faster than another. But how is a strategy for the mental, moral and physical destruction of an opponent applicable to software development? Chet. Richards provides his thoughts:

“I am not saying that the specific tactics of maneuver warfare—or any other form of warfare—apply to business. However, I am claiming that Boyd’s underlying strategy – the use of time as a shaping and exploiting mechanism and the emphasis on a culture/organizational climate that makes this possible—apply equally well to both” [10 p.132].

Still, does a competition based decision model apply to software development? In our opinion the answer is yes. First, while a software development team may not consider itself in competition with an adversary, rapid unanticipated changes emerging from the environment can have the same disorienting effect on a team as an explicit adversary. An inability to quickly respond to changes will lead to confusion and disorder in a software development team as surely as if that team were in competition with an adversary. To paraphrase Boyd, our organization’s objective is to survive on its own terms [9 p.10].

Second, the OODA loop is about getting into a position of advantage in an uncertain environment. When in a position of advantage we have more degrees of freedom to take the initiative and shape our environment. Even with this adversarial philosophy the OODA is useful for us because in the seemingly co-operative game of software development we’re still seeking advantage over the environment, to cope with change, and shape the environment. At its most basic, the OODA model is about groups of people working harmoniously together to accomplish a common purpose—even though everything is collapsing around them. This describes a lot of human activity, including all sorts of development programs [15].

Finally, lest we forgot, software development is commerce and commerce is about competition. Most software development activity takes place in corporations that operate in highly competitive environments. Our competitors are often directly or indirectly contributors to uncertainty. It is the reality of that competitive environment, uncertainty, fluidity, time, ambiguous requirements that created in the interest in agile software development.

6. The OODA Loop’s Implications for the Agile Community

There are several implications for the agile community resulting from Colonel Boyd’s work:

First, agility is a time based strategy for operational success and is not based on size. A project is agile if it is able to execute its reorienting and action-taking cycle faster than the changes occurring in its environment. So in pilot’s language, it is always ahead of the power curve. It is worth remembering that Boyd

began development of the OODA loop as a description of a fighter pilot’s decision making process and developed the OODA loop into a general theory of competition and maneuver warfare. The US Marine Corps directly acknowledge their doctrines are derived from Boyd [16] and companies have adopted parts of Boyd’s strategies, though some without acknowledging him [17].

Second, agility is a relative concept, not an absolute concept. An agile project is one that works its OODA loop more quickly than changes can occur in the environment. A large slow moving project is agile if it can respond and shape changes in its environment in a timely manner. A small quick moving project may not be agile if it cannot stay ahead of and shape changes in its environment. The agile methodologies are designed for operating in rapidly changing uncertain environments. But this does not mean that larger methodologies cannot be agile if they correctly match the environment.

Third, agility is an important attribute for project success, large or small. If we cannot execute our OODA loops more quickly than the changes are occurring in our environment, then the result will probably be that our project will get further and further behind, internal purpose and harmony will break down under the stress, and failure is likely.

Fourth, agility depends on organizational culture. The talents that reduce our cycle time—unity and trust, skill and experience, mission, and focus—are cultural. Agile methodologies were created with the intention of unleashing these talents but a methodology cannot compensate for the lack of these talents in an organization. This is why it can say agile on the box, but not feel like agile. Furthermore, an organization using a “plan driven” methodology can be agile, if the plan and other prescribed methodological artifacts do not disconnect the team from its environment. Methods and tools are secondary to the development of a team because a smooth functioning, agile, team will take the initiative to discover or invent these for itself. To paraphrase an old US election slogan “it’s the people, stupid!”

Finally Boyd’s work further reminds us that our recent discovery of agility only means we are late to the party.. The benefit for us is the massive body of knowledge and experience for us to research and learn how to apply to software development.

7. Lessons and Research Opportunities

Boyd’s OODA loop is a time based definition of agility and shows agility depends on organizational culture and climate, and not on size or tools and processes. This cultural dependency explains why a

project using an agile methodology may not feel agile, and why a so called plan driven project may feel agile. Agility becomes the study of how culture can provide a competitive advantage. A high profile example of a company using culture to competitive advantage is Southwest Airlines [18].

The principles for operational success—mutual trust, skill, intent, and vision—describe the necessary talents for an agile organization. Unfortunately, these are generally not talents that emerge spontaneously in an organization. These are talents that must be learned and carefully cultivated. The military is famous for its ability to create an effective team from a group of strangers. In a comparison between agile software development and war fighting, Adolph noted the importance the military and specifically the United States Marine Corps place on training to develop these talents. [19] However, most software developers and most software development organizations are probably not inclined to implement the strict discipline of military training.

Fortunately there are other organizations we can study to learn how to develop effective teams. One industry that may be a rich source of analogues is air transport. Early in the 1980s several reports suggested that aviation accidents blamed on pilot error were more likely failures in team communication and coordination than deficiencies in “stick and rudder” proficiency [20 p.7]. This should resonate with many software project failure survivors, because in many of our personal experiences project failure more often results from failure of the team to work together than from a lack of technical skill.

As a result of these reports airlines developed training programs to teach team skills to flight crews frequently known in the industry as Cockpit Resource Management (CRM). While initially derided by pilots as “charm school”, a number of pilots have suggested their CRM training was invaluable in preventing a major airline tragedy. One research direction we are therefore investigating is possibility of creating a similar program for software teams, perhaps a Developer Resource Management Program

8. Summary

Colonel John Boyd provides us with a time based definition of agility and further shows us that agility is a cultural trait. A key lesson from Colonel Boyd’s theory is the importance of team culture to quickening the decision cycle. Colonel Boyd summed this lesson up in his admonition “People, ideas, hardware – in that order” [8]. We must not let our fascination with processes and tools relegate the development of interpersonal and teams skills to the back burner. Like

flight crews, our technical skills do not guarantee success and we must learn to take full advantage of all resources available to us.

We believe Colonel Boyd’s theories are applicable to software development and explain why an agile project may not feel agile and why a so called “plan driven” project can feel agile. We also asserted that agility is an important attribute for a successful project since it is the most effective way of dealing with uncertainty and avoiding or if necessary responding effectively to crises. Success through agility is a strategy employed by many companies and therefore there are many resources and lessons learned the software community can draw on to help us develop our strategies. We are beginning to investigate the Cockpit Resource Management programs used to train pilots to work effectively in teams as the possible basis for a similar program for software developers.

10. References

- [1] K. Beck and C. Andres, *Extreme Programming Explained*: Addison-Wesley Professional, 2004.
- [2] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Upper Saddle River, N.J.: Prentice Hall, 2002.
- [3] P. Mcbreen, "Pretending to be Agile," 2002, available at www.informit.com/articles/article.asp?p=25913&seqNum=1&rl=1
- [4] A. Cockburn, "Are Iterations Hazardous to Your Project," *Humans and Technology*, 2005.
- [5] P. Kruchten, "Agility with the RUP," in *The Rational Edge*, 2002, available at www.ibm.com/developerworks/rational/library/content/RationalEdge/jan02/AgilitywiththeRUPJan02.pdf
- [6] M. Hirsch, "Making RUP Agile," 2002, available at www.agilealliance.com/articles/hirschmichaelmakingrup/file
- [7] S. Ambler, "The Agile Unified Process," 2005, available at www.ambysoft.com/unifiedprocess/agileUP.html
- [8] R. Coram, *Boyd: The Fighter Pilot Who Changed the Art of War*. New York New York: Back Bay Books 2002.
- [9] J. Boyd, "Patterns of Conflict," 1986, available at www.d-n-i.net/boyd/pdf/poc.pdf
- [10] C. Richards, *Certain to Win: The Strategy of John Boyd, Applied to Business*: Xlibris Corporation, 2004.
- [11] J. Boyd, "Organic Design for Command and Control," 1987, available at <http://www.d-n-i.net/boyd/pdf/c&c.pdf>

- [12] C. Richards, "Certain to Win: John Boyd's Strategy in the 21st Century," J. Addams & Partners, Inc., 2005.
- [13] G. Klein, *Sources of Power: How People Make Decisions*. London, England: The MIT Press, 1998.
- [14] C. v. Clausewitz, *On War*. London: Penguin Books, 1982.
- [15] C. Richards, private communications, 2006
- [16] USMC, "MDCP 6 Command and Control," U. S. M. Corps, Ed., 1996.
- [17] J. L. Bower and T. M. Hout, "Fast Cycle Capability for Competitive Power," *Harvard Business Review*, 1988.
- [18] C. O' Reilly and J. Pfeffer, "Southwest Airlines: Using Human Resources for Competitive Advantage(A)," *Graduate School of Business Stanford University HR-1A*, 1995.
- [19] S. Adolph, "Are we ready to be unleashed? A comparative analysis between agile software development and war fighting," presented at Agile Conference, 2005. Proceedings Denver, 2005.
- [20] E. L. Wiener, B. G. Kanki, and R. L. Helmreich, "Cockpit Resource Management." San Diego California: Academic Press, 1993.