

# Symbolic Innovation in Agile Transformations

Doug Rose

Syracuse University  
School of Information Studies  
Syracuse, USA  
dbrose@syr.edu

**Abstract**—Much of the empirical research on agile transformations concentrates on the success factors for effective change management. Although these factors are essential in establishing beneficial norms and practices, the existing research does not adequately address the importance of language in distinguishing the current plan-driven processes from their agile counterparts. The labels in agile and plan-driven processes represent practices that are well established. Accurate process labeling is the foundation for creating any inter-organizational lessons. This paper establishes the link between linguistic manipulation and how it might lead to a form of symbolic innovation that can impede a greater understanding of transformational challenges. During an agile transformation some organizational actors re-label deterministic plan-driven processes using agile language. The symbolic innovation leads to no improvement and as a result is more easily abandoned without meaningful process change.

**Keywords**—*agile, transformation, re-labeling, language, software development, project management, symbolic innovation*

## I. INTRODUCTION

For over four decades, software development has been rooted in deterministic planning processes. Starting in the 1970s software development embraced contemporary manufacturing processes. The standard manufacturing phases included planning, requirements gathering, architecture, implementation and testing. In the early 1990s many organizations took an interest in Toyota's lean manufacturing. Lean manufacturing centered on continuous improvement and value-driven delivery. Lean principles improved delivery through adaptive planning and emphasized people over process. Software development had several characteristics that mirrored lean principles: a team of highly skilled engineers creates software and valuable changes are part of continuous improvement. In 2001, software developers established a framework to apply lean and other lightweight development methods. They called the framework "agile development" as a way to emphasize the lightweight adaptable approach.

Agile is a relative newcomer for organizations and most structures and norms continue to support plan-driven development. The phases of planning, implementation and testing were original templates for common organizational roles. Detailed plans are a part of the Project Management Office (PMO). Software testing is the responsibility of the Quality Assurance department. In plan-driven projects, the business requirements documentation (BRD) is the predetermined map for the project. Agile transformations

subvert many of these norms and structures. Agile combines project phases into a repeatable iteration. Agile developers self-organize and make incremental adjustments that were originally part of a project control process.

Many organizations opt to incrementally embrace agile principles instead of a wholesale abandonment of their established control practices. This led to common techniques that allowed organizations to explore agility and still leverage their current processes. One such technique is a hybrid approach. The hybrid approach is an overlapping of the plan-driven and agile development frameworks. The hybrid approach can lead to confusion about the language and purpose of agile. Some plan-driven processes transform into re-labeled agile processes without much meaningful innovation. The re-labeled processes cause significant organizational inefficiency. Organizational stakeholders looking for improvement find little impact. The teams participate in familiar development practices with unfamiliar language. The agile transformation becomes a symbolic innovation. The language changes but the process remains the same. This stifles an already challenging transformation process.

The agile principles of freedom, accountability and empowerment are not always consistent with plan-driven processes that favor control, compliance and management. This can make well-established organizational actors suspicious of agile transformations. The bargain of releasing control for adaptability is essential for building a self-managed team. This exchange might not be an easy arrangement for managers who have established compliance roles. The process relabeling allows these roles to continue unabated in their original duties.

This paper attempts to illustrate how symbolic innovation can hinder agile transformations. The hybrid approach is designed to ease an organization into agility, but may in fact cause too much language overlap that ultimately protects the status quo.

## II. LITERATURE TO DATE

Software development is an arduous process that requires a defined project management process (Charette, 2005). The most widely used software development frameworks are the plan-driven and the agile approach (Paulson, 2001). Initially, practitioners drove theory on agile development (Conboy, 2009). Plan-driven development produced high failure rates (Standish Group International, 1995) and lacked flexibility for rapid changes (Boehm and Turner, 2003). The term "agile" was coined as part of the Manifesto for Agile Software

Development ([www.agilemanifesto.org](http://www.agilemanifesto.org)). The manifesto is the original framework that defined a software development approach that was lightweight and adaptive. The Manifesto listed four values: individuals and interactions over processes and tools; working software over comprehensive documentation; customer collaboration over contract negotiation; and responding to change over following a plan. The values are a pendulum that pulls back on established plan-driven software development processes. The agile values imply that plan-driven development overemphasizes processes and tools, comprehensive documentation, contract negotiation and following a plan. Agile is an umbrella term that encompasses several lightweight development frameworks. The common trait of agile is the ability to adapt to rapid change, with a short customer feedback cycle and an emphasis on value over delivery.

Scholarship on the agile framework has been limited (Dybå and Dingsøy, 2008). Some of the literature has focused on agile development practices (Williams, 2010), other empirical research describes extending the framework (Agerfalk, Fitzgerald and Slaughter, 2009) and creating derivatives. There is work on how the derivatives impact the original framework (West and Grant, 2010). Much of the research addresses the challenges with adopting agile process in plan-driven environments. The agile transformation (Hirsch, 2005) replaces some or all of the plan-driven processes with agile processes (Laanti, Outi and Abrahamsson, 2011) or creates a hybrid approach (Fitzgerald, Hartnett and Conboy, 2006; Hayata and Han, 2011; Petersen and Wohlin, 2010)

The work on organizational change and linguistic manipulation is more diverse. The relevant research centers on how organizations going through a process change often create members who resist the change. They are the “organizational antibodies” that protect the current processes from foreign ideas (Kelley, 2005; Oster, 2008a; Oster, 2008b; von Krogh, Ichijo and Nonaka, 2000). The more radical the ideas appear, the more intense the response (Davila, Epstein and Shelton, 2006).

Organizational antibodies may use linguistic manipulation as a form of symbolic innovation. In cognitive terminology, a symbolic innovation is when a process acquires new intangible attributes even though the substance of the process remains unchanged (Hirschman, 1982). The name of the process changes to match the proposed innovation, so the relabeling supplants innovation. According to Asya (2013) the process relabeling is an “indirect method of expression of communicative intention [that] presupposes usage of language forms to express illocution force not connected with their direct linguistic meaning” (P.3). The language and terminology communicate a change when the opposite is true. It is a process re-labeling without a process retooling. Savaneviciene and Stankeviciute (2013) point out that “personnel management” replaced “human resource management” and this led to a perceived process transformation, but the change was primarily a re-labeling.

### III. CONCEPTUAL FRAMEWORK

Whether or not agile is an improved development framework is not the focus of this research. What is essential is that agile development is distinguishable from plan-driven development. Many organizations are drawn to agile’s allure of adaptive incremental delivery (Dybå and Dingsøy, 2008). Plan-driven development often struggles with changing development priorities (Pikkarainen, Salo, Kuusela, and Abrahamsson, 2012) but is often a better choice for projects that require repeatability, scalability or high security (Petersen, Wohlin and Baca, 2009). The mixture of benefits leads to agile transformations with incremental adoption of some agile processes.

The aim of this research is to show that organizations use symbolic innovation as part of the transformation. The agile framework has different roles, planning processes, artifacts and workspace constraints. Most organizations have a substantial investment in plan-based development processes. The re-labeling over retooling may be the preferred approach when organizational antibodies are averse to the change.

The motivation for symbolic innovation is threefold. The re-labeling makes the process more familiar. The labels make the process less threatening to antibodies that have sunk cost in the status quo. It also might quell risk aversion (Slovic, Peters, Finucane and MacGregor, 2005). The re-labeling might be particularly comforting for anyone who fears their role in the organization is threatened by the transformation (Oster and Gandolfi, 2008).

Organizations may unintentionally re-label as a by-product of the hybrid approach. The overlapping of two approaches will mix labels and facilitate symbolic innovation. The symbolic innovation will re-label plan-driven processes with their contradictory agile counterparts. This changing process maintains the original qualities of its plan-driven origins even when they are in opposition to the well-established values of the agile manifesto. Unchanged processes, that predictably provide no discernible benefit, mire the overall goals of the agile transformation.

### IV. METHODOLOGY

My inquiry was guided by two questions. First, I wanted to determine if symbolic innovation is a significant factor in agile transformations. Second, I wanted to ascertain if the process relabeling conflicts with the plain-language values in the agile manifesto and the well established agile principles of adaptation and efficiency (Dybå and Dingsøy, 2008). My field research recorded the agile process and interviewed the team members about their understanding of the artifacts and practices involved in the process.

#### A. Research Site

The research site is the headquarters of a large retailer in the United States. The company was founded in the late 1970s and currently has several hundred thousand employees with annual revenue of tens of billions of dollars. The project was a web-based software application that consolidated the view of several applications used by the customer service call center. In

the current system, the customer support representatives or “CSRs” would keep several applications open to search for inventory, change orders, and create orders. Management tasked the chief executive for human resources with consolidating the application. The human resource executive worked closely with the chief information officer. The budget was \$12 million with an 18-month timeframe.

The organizational makeup included several dozen project managers who worked closely with a team of business analysts. The teams included a mix of contracted and staff positions. The majority of software development was offsite and contracted to several large staffing companies.

I chose the organization for several reasons. First, the project was well funded and the organization was committed to an agile transformation. Second, the organization had a mid-level agile “champion.” This person was the change agent for the agile transformation and had a rudimentary knowledge of the framework. Third, the project heavily invested in plan-driven software development. Fourth, the organization was receptive to the research, provided the details about the company were omitted. For this research paper I will use the name HomeMart as a pseudonym.

#### B. Data Collection and Analysis

Direct observation occurred from June to November 2013. The author observed regular work practices for the projects teams full-time during normal business hours. Direct observation formed the bulk of this field research, including observation of the customer service call center and visits to the out-of-state office that developed the agile transformation guidelines. The author conducted informal interviews with the employees and contractors. Pictures and notes were taken at the meetings and of agile artifacts (task boards, shared workspace, meeting rooms). The transformation used Scrum as the agile development framework and so the language and terms will reflect that choice.

### V. CASE FINDINGS

In reporting my findings, I created three groupings of symbolic innovations: agile roles, agile planning, and the agile workspace. Each process relabeling was inconsistent with a plain-language reading of the manifesto and lean principles. The agile manifesto has been interpreted and clarified through succeeding literature. The plain language view of the values will potentially remove a layer of abstraction.

#### A. Symbolic innovation of agile roles

Agile teams are self-organized, aware of customer value, and provide daily feedback. Typically project managers and business analysts perform these tasks in plan-driven software development projects. Scrum defines three roles: the developer, the ScrumMaster, and the customer representative or Product Owner. The ScrumMaster focuses on removing obstacles. The ScrumMaster role is designed to “master” Scrum and ensure that the team is correctly following the framework. In other agile frameworks this role is often called a “coach.”

Project managers primarily deal with compliance with the project plan. They ensure that the project is delivered with the scope of deliverables, on time, within budget and with a predetermined quality.

HomeMart had long-term contractors who previously acted as both development leads and project managers. The same mid-level management staff relabeled themselves as ScrumMasters. The symbolic transformation was limited to relabeling. The role of the ScrumMaster is to act as a team facilitator without traditional management authority. The role encourages the team to reach a sustainable level of self-organization. The role at HomeMart was to “manage the team” and “determine outcomes.” This was much more closely aligned with a project manager. The ScrumMaster was focused on constraints and control. They asked the team questions centered on schedule and delivery. The agile champion for the project affirmed this role and added that ScrumMasters were accountable to deliver the software milestones.

*“We are using a hybrid approach so we still need our ScrumMasters to maintain a level of accountability.” [Agile Champion]*

The ScrumMaster also changed the purpose of agile meetings to reflect their project manager role. The daily stand-up meeting was a re-labeled status meeting. In an agile daily stand-up meeting the self-managed team updates other team members on their progress and obstacles. The ScrumMaster records the obstacles as a way to ensure the team focuses on development. The format is that each team member updates other developers with the answers to three questions: “What did I do yesterday?” “What am I doing today?” and “Do I have any obstacles?” The daily stand-up meeting typically lasts between fifteen and twenty minutes and is held first thing in the morning at 9 AM.

The HomeMart daily stand-up meeting was led by the ScrumMaster. The ScrumMaster would gesture to each developer and ask the questions directly:

*“What did you do yesterday? What will you do today? What obstacles are in your way?” [ScrumMaster]*

The developers looked only at the ScrumMaster and committed to delivery dates. The meeting didn’t begin until the ScrumMaster was attending. If the ScrumMaster wasn’t available the meeting was cancelled.

There was also a daily stand-up meeting at the program level. The meeting was listed in the group calendar as “Daily Stand-up Meeting.” This meeting was held in a large conference room with a large oval table in the center of the room. The program manager sat in the middle of the table and organized the meeting from the central position. The meeting also connected through a videoconferencing bridge to a large office off-site. The meeting began at 9:30 AM Monday through Friday and would start when the program manager sat at the desk. Everyone was sitting throughout the meeting and it was scheduled until 11:00 AM. If the program manager didn’t have any questions they would reach out to the team to make sure that all the allotted time was used.

*“We have about 20 minutes left in our Stand-up meeting. So let’s go around the room and everyone give us a status update.” [Program Manager]*

Although the format was different, the purpose of the meeting is consistent. The meeting organizer reports the status of deliverables to upper management. The status update and approval process is a likely contradiction of the agile value of “individuals and interactions over processes and tools.”

### *B. Symbolic innovation of agile planning*

Responding to changes is more important than following a project plan. Agile stays true to its lean manufacturing heritage by minimizing the waste in producing low value documentation (Petersen and Wohlin, 2010). Working software provides the greatest value to the customer. That is why the agile framework strives to produce “barely sufficient” (Highsmith, 2009) documentation.

Plan-driven projects depend on clear customer requirements at the beginning of the project. If those requirements change then a project manager applies a change request or “CR.” HomeMart created Business Requirements Documents or BRDs for the CSR consolidation project. The first six months of the project were set aside for the *planning sprint*. Once the BRDs were completed, the software requirements were copied into user stories and placed into the product backlog. Once the stories were in the product backlog they went through the same change control process. An approved CR would impact both the BRD and the user stories in the product backlog.

*“The stories won’t change unless they come from an approved CR.” [Agile Champion]*

At the Sprint planning meeting, the ScrumMaster would pull approved CRs from the BRD and add them to the sprint backlog. The team would estimate the stories to verify the project was on schedule. The ScrumMaster would work toward “maximum velocity” and deliver the stories by milestone dates established in the BRD.

### *C. Symbolic innovation of the agile workspace*

The shared workspace in an agile team is one of the material affordances that communicate freedom and adaptability (Wagner, Newell and Ramiller, 2013). An agile shared workspace is designed to foster communication and collaboration. This usually includes an open space to facilitate impromptu discussion. The purpose of an agile shared workspace is to remove obstacles to interpersonal communication.

HomeMart was housed in a multilevel office building with several towers. The majority of the floor space contained interlocked grey cubicles and the floor’s surrounding white walls had offices with glass doors. Some of the staff had nameplates on the end of each “cube row.” Many of the contracted staff made makeshift paper nameplates and some occupied cubicles were undesignated. The cubicles had a sophisticated numbering system. The location numbers were based on the floor, the row and cardinal direction. For meetings

it was not uncommon to use cubical numbers as meeting points. So you might hear a staff member say, “please meet me at my cubicle 19DSW2045.” The layout and material affordances communicate that each person is a discrete part of a larger organization. The cubicle walls afford each person an isolated workspace to focus on their own work.

At HomeMart, the agile team was distributed in cubicles throughout the organization. Team members worked on different floors, in different buildings and in different cities. The program manager didn’t create a shared agile space for all the team members because of the functional makeup of the organization.

*“We have too many resources to take down the cubicles. So the agile shared workspace is next to their functional managers” [Program Manager]*

The shared workspace was a space that had one or more agile team members. There were several team members who worked for different functional managers so the agile team had several shared workspaces. HomeMart re-labeled the clusters of cubicles agile shared workspaces.

## VI. DISCUSSION

At HomeMart, the re-labeling over retooling led to peculiar practices. They had daily stand-up meetings where no one stood up. Instead of mastering Scrum, the ScrumMaster ignored key tenets of the framework. The team worked in cubicles that were paradoxically relabeled the agile shared workspace. The labels of the processes and artifacts contradicted their definitional interpretation. The process labels were more coherent when reverted to their original plan driven language. The stand-up meeting should be a project status meeting. The ScrumMaster should be a project manager. The isolated cubicles should be a private workspace.

It harms the agile transformation when agile labels litter plan-driven projects. Labeling acts as a departure point for organizational transformation. Organizational change is a journey. Like most journeys the route depends on the position and the destination. Without these two points there is no measurement of progress. The path to innovation is not navigable when labels do not accurately reflect either the status quo or the transformed state.

The motivation to label plan-driven process with agile terms may be innocent or obstructionist. In either case, the result is the same: confusion and no discernible improvement. If a fast food restaurant re-labeled their triple-cheese burger a low-fat sandwich it would confuse both the dieters and render the diet meaningless. Diligent dieters would watch their waistlines grow and eventually declare the process a failure. Agile transformations with re-labeled processes follow the same fate. HomeMart eventually abandoned their agile transformation after they found no discernible benefit. In a follow-up interview, the agile champion said that agile was not ready for “prime-time.” From their perspective the conclusion is understandable. The ScrumMasters organized stand-ups. The team used a product backlog and created user stories. The labels created the rhythm and cadence of an agile

transformation. The project eventually reverted to its appropriate labels and followed the plan-driven approach until completion. The ScrumMasters once again became project managers and the majority of the developers were re-staffed. The confusion and churn caused the project to run over budget and potentially doomed future transformations.

## VII. FUTURE RESEARCH

This paper focused on the *how* of agile process re-labeling and not the *why*. An opportunity for further research is hinted at throughout the case. Was the process re-labeling intentional or a byproduct of the hybrid approach? Within the case, there was some evidence of a mixture. The hybrid approach caused confusion that was left uncorrected by project managers. At HomeMart, the project managers and business analysts were the transformation experts. They were the staff that attended the formal agile training. It was their responsibility to retrain the organization. At HomeMart, they neglected these duties.

It is understandable that some project managers have a hard time accepting some of agile's key differentiators. A project manager might not readily embrace self-management after a career of balancing constraints. There was some evidence at HomeMart that the project managers concluded that agile was simply a rewording of what they've been doing all along. "Meet the new boss. Same as the old boss." The ScrumMaster's duty to remove obstacles might be deceptively familiar to long-term project managers. A follow-up study on what longtime project managers retain from agile training would help answer this question.

This research is also limited to the normal constraints surrounding a single case study. Most organizations transition to agile using an incremental approach. It is plausible to assume that process re-labeling is widespread. Further research is needed to compare multiple organizations attempting agile transformations.

## REFERENCES

- [1] Agerfalk, P. J., Fitzgerald, B. and Slaughter, S. A. 2009. "Flexible and distributed information systems development: state of the art and research challenges," *Information Systems Research*, 20, 3, 317-328
- [2] Asya, A. 2013. "Linguistic manipulation: Definition and Types," *International Journal of Cognitive Research in Science, Engineering and Education*, 1(2).
- [3] Boehm, B., & Turner, R. 2003. "Observations on balancing discipline and agility," In *Agile Development Conference, 2003. ADC 2003. Proceedings of the* (pp. 32-39). IEEE.
- [4] Charette, R. N. 2005. "Why software fails," *IEEE spectrum*, 42(9), 36.
- [5] Conboy, K. 2009. "Agility from first principles: reconstructing the concept of agility in information systems development," *Information Systems Research*, 20(3), 329-354.
- [6] Davila, T., Epstein, M., & Shelton, R. 2006. *Making innovation work*, Upper Saddle River, NJ: Wharton School Publishing.
- [7] Dybå, T. and Dingsøyr, T. 2008. "Empirical studies of agile software development: A systematic review," *Information and Software Technology*, 50, 9-10, 833-859.
- [8] Fitzgerald, B., Hartnett, G. and Conboy, K. 2006. "Customising agile methods to software practices at Intel Shannon," *European Journal of Information Systems*, 15, 200-213.
- [9] Fowler, M., & Highsmith, J. 2001. "The agile manifesto. *Software Development*," 9(8), 28-35.
- [10] Hayata, T., & Han, J. 2011. "A hybrid model for IT project with Scrum. In *Service Operations, Logistics, and Informatics (SOLI)*," 2011 IEEE International Conference on (pp. 285-290). IEEE.
- [11] Highsmith, J. 2009. *Agile project management: creating innovative products*, Pearson Education.
- [12] Hirsch, M. 2005. "Moving from a plan driven culture to agile development," In *International Conference on Software Engineering* (Vol. 27, p. 38).
- [13] Hirschman, E. C., 1982. "Symbolism and technology as sources for the generation of innovations," *Advances in Consumer Research*, 9(1), 537-541.
- [14] Kelley, T. 2005. *The ten faces of innovation*, New York: Currency Doubleday.
- [15] Laanti, M., Outi, S. and Abrahamsson, P. 2011. "Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions of agile transformation," *Information and Software Technology*, 53, 3, 276-286.
- [16] Oster, G. 2008. "Derailing design thinking," *International Journal of Leadership Studies*. 4(1), 107-115.
- [17] Oster, G. 2008. "Divining the need: compensatory behavior of customers," *Regent Global Business Review*, 2(2), 14-18.
- [18] Oster, G., & Gandolfi, F. 2008. "Innovation during an era of downsizing," *Review of International Comparative Management*, 9(5), 125-145.
- [19] Paulson, L. D. 2001. "Adapting methodologies for doing software right," *IT Professional*, 3(4), 13-15.
- [20] Petersen, K., & Wohlin, C. 2010. "The effect of moving from a plan-driven to an incremental software development approach with agile practices," *Empirical Software Engineering*, 15(6), 654-693.
- [21] Petersen, K., Wohlin, C., & Baca, D. 2009. "The waterfall model in large-scale development," In *Product-Focused Software Process Improvement* (pp. 386-400). Springer Berlin Heidelberg.
- [22] Pikkarainen, M., Salo, O., Kuusela, R., & Abrahamsson, P. 2012. "Strengths and barriers behind the successful agile deployment—insights from the three software intensive companies in Finland," *Empirical software engineering*, 17(6), 675-702.
- [23] Savaneviciene, A., Stankeviciute, Z. 2013. "Relabeling or new approach: Theoretical insights regarding personnel management and human resource management," *Engineering Economics*, 24(3), 234-243.
- [24] Slovic, P., & Peters, E., & Finucane, M.L., & MacGregor, D.G. 2005. "Affect, Risk, and Decision Making," *Health Psychology*, 24, S35-S40.
- [25] Standish Group International. 1995. *Chaos Report*
- [26] von Krogh, G., Ichijo, K., & Nonaka, I. 2000. *Enabling knowledge creation*. Oxford, UK: Oxford Press.
- [27] Wagner, E. L., Newell, S., & Ramiller, N. 2013. "The Structuring of Freedom in Agile Development,"
- [28] West, D., & Grant, T. 2010. "Agile development: Mainstream adoption has changed agility," *Forrester Research*.
- [29] Williams, L. 2010. "Agile software development methodologies and practices," *Advances in Computers*, 80, 1-44.