

Portland State University

PDXScholar

Business Faculty Publications and
Presentations

The School of Business

12-1-2018

From Public Ideology to Socio-material Reproduction of Agile Principles: The Case of Pivotal Labs

Erica L. Wagner

Portland State University

Sue Newell

The University of Warwick

Neil Ramiller

Jeanne Enders

Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/busadmin_fac



Part of the [Business Commons](#)

Let us know how access to this document benefits you.

Citation Details

Wagner, E. L., Newell, S., Ramiller, N., & Enders, J. (2018). From public ideology to socio-material reproduction of agile principles: The case of pivotal labs. *Information & Organization*, 28(4), 192–210. <https://doi.org/10.1016/j.infoandorg.2018.10.002>

This Post-Print is brought to you for free and open access. It has been accepted for inclusion in Business Faculty Publications and Presentations by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

FROM PUBLIC IDEOLOGY TO SOCIO-MATERIAL REPRODUCTION OF AGILE PRINCIPLES: THE CASE OF PIVOTAL LABS

ABSTRACT

This paper focuses on how regular invocation of a public idea helps sustain local work practices. Existing research acknowledges that there is a recursive relationship between the local practices and the public ideas, with local practices leading to the adaption of public ideas over time. There is little study of how the public idea sustains local practices over the longer term – seeing translation of the broad ideology as an ongoing process. Through a case study and web-based analysis of Pivotal Labs, an agile software development company, we contribute to socio-material translation studies demonstrating the manifestation of the public idea in socio-material practices. We also introduce the concept of a *collective cognitive bridge* that supports the continued relevance of the public idea as a prop for these socio-material practices.

Keywords: socio-materiality, translation, agile software development, qualitative research, web-based research, field study

1. INTRODUCTION

Over my four-year tenure at Pivotal Labs I've heard it a lot: 'You guys really drink the Kool-Aid around here, don't you' and I'd shake it off with a joke ... I thought I knew what they were getting at, we're pretty rigorous about things that are radically different from the way most software developers work—[test driven development], pair programming, and a common start time to name a few. But it takes more than common practices to build a culture that is so strong that it consistently elicits cult analogies. [WR 5/4/13]

This employee blog post about Pivotal Labs (PL), an agile software development consultancy, is consistent with how others describe the company's culture. PL is known for its systematic and consistent adoption of agile methodology in its creation of software as well as its adherence to a set of defined work practices. In this article, we focus on the role that public ideology serves for members of PL and those within the consultancy's orbit as work was done, projects completed, and reputations developed. A public ideology refers to the belief system of a social group which embraces particular values and provides guiding principles for individual and collective action (Johnston and Oliver 2000 within Barrett et al 2013). The belief system is manifest in ideological discourse, which in the case discussed in this paper centers on a tract known as the Agile Manifesto. The Agile Manifesto is a seminal document that refers to a belief system for agile software development. The Manifesto was written in 2001 by 17 software developers in response to rigid and sequential approaches to development (such as Big Design Up Front (BDUF), Waterfall). The Manifesto itself says nothing about methods for realizing these values. It is philosophical and ideological in nature:

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value: Individuals and interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan. That is, while there is value in the items on the right, we value the items on the left more.”(Beck et al. 2001)

At the time of this writing, the Manifesto is seen as the core set of beliefs of the agile development trend and has been translated into 60 languages and signed by more than 20,000 people up to July 2016 when it was closed to signatures.¹

In this paper we show that the Manifesto serves as a talisman guiding and helping maintain work practices within PL consultancy. When there are questions related to performing work, the Manifesto informs the answer by grounding employees and clients in its tenets and reminding them of what PL aims to do and how. After all, organizations rarely acquire new practices through acts of pure invention. Instead, they draw on the prevalent wisdom created by, conveyed in, and legitimized through broader industry discourses. In this paper we focus on the link between the specific ideology of agile software development – and the realization of local socio-material work practices that are sustained over time.

We draw on the conceptual apparatus of socio-materiality to help us address the link between public ideology and local socio-material practice. We follow Leonardi (2012) in defining socio-materiality as the “enactment of a particular set of activities that meld materiality with institutions, norms, discourses, and all other phenomena we typically define as social” (34). This

¹ All signatories and their comments are found at AgileManifesto.org

melding is referred to as “imbrication” signifying an ‘interlocking sequence of different types of agencies that produce a visible pattern’ (ibid). This interpretation of the socio-materiality of information technology helps us see how the materiality of technologies enable and constrain certain work practices that acquire meaning and practical significance in the context of use even as those practices are enabled by—indeed, often unimaginable without—those technologies (Leonardi, 2013). Practices, for their part, are not just whatever people want to do on this or that occasion with the tools at hand; instead, they are instantiated in regularized patterns that express underlying organizational culture/norms. Most studies of socio-material practices have, to date, offered localized accounts of the execution of existing practices. Those studies have not paid much attention either to how such practices originate or to the challenge of maintaining them over time. This is where the role of larger public discourses in introducing and reproducing favored practices and their associated socio-material agencies becomes relevant. Scholarship has begun to consider the organizational appropriation of normative public discourse into organizational practices (Wagner, Scott and Galliers, 2006).

We aim to contribute to this body of knowledge by focusing on understanding the mechanisms and processes of translation and continued enactment of concepts from institutional environments into local practice. To date, IS literature has looked at the processes of translation but the point here is that it is also the ongoing enactment of these practices after the initial translation that has not been thoroughly considered. As such, our research is guided by the question: how does a public ideology play a role in the local reproductions of work practices? In order to explore this question, we conducted an in-depth field study of agile software developers working at PL. This field study was coupled with the collection of publically available online data sources. We selected PL as the field study site for several reasons. First, it is recognized by the

software development industry as a leader in agile development. Second, the organization develops software exclusively using an agile approach. Third, the organization was receptive to the research and provided access to its customers, employees, and executives at multiple locations.

We study the relationship between organizational work and public ideology by examining how and why social norms, policies, and communication patterns as well as physical and digital materiality intertwine to become sustained socio-material work practices. We discuss how the Agile Manifesto operated in the context of PL's method for developing software. We investigate how the Manifesto facilitated the translation and ongoing enactment of concepts from the institutional environment into practice that allowed the ongoing realization of agile principles. We highlight specifically, the pair programming (PP) practice that serves as a medium for how the imbrication of socio-material agencies is sustained by the support of ideology. Together these local and more macro influences allow for the realization of agile principles through the daily work practices of software developers.

In preparation for investigating this phenomenon through the PL case, the paper proceeds in section 2 with a literature review of two intersecting domains: translation and socio-materiality. These domains form the conceptual foundations of the paper and aid in the analysis of the PL case study. Against this background, the authors present the research design and the PL case in section 3. In section 4 we present the analysis of our findings and show how a public idea sustains local practices over the longer term. In section 5, we focus on the manifestation of the public idea in socio-material practices but also the mechanisms that support the continued relevance of the public idea as props for these practices. The paper concludes in section 6 and we suggest areas for future research.

2. CONCEPTUAL FOUNDATION

The theoretical underpinnings of this paper center on the relationship between a public ideology and its local instantiation - the *travel of ideas* and their requisite translation and ongoing enactment in particular organizational settings (Nielsen et al., 2014). By extension, the concept of translation involves the local adaptation and incorporation of innovative public ideas into the work practices of the organization (Nelson and Winter, 1982). With an interest in understanding the organizational development of IT-infused practices, this inquiry examines how the instantiation and ongoing reproduction of the public idea is necessarily socio-material in its realization (Leonardi, 2011).

2.1. The Travel of Ideas and Their Organizational Translation

Information Systems scholars have long appreciated that organizational innovation depends as much (or more) on the diffusion of new ideas across institutional fields as it does on local invention (Swanson, 1994; Rogers, 2003). Grand public ideologies or theories for innovating with IT evolve through discourse in larger communities which then become diffused and adopted by individual organizations (Swanson and Ramiller, 1997). Diffusion of a new idea to a particular organization will involve the adoption of material instantiations of the idea (technological and other material artifacts) alongside changes to the social arrangements of work. This is not a straight-forward process, however, because the IT ideas or theories must be reinterpreted and implemented in ways that make sense in the particular organizational setting. Thus, Nielsen et al., (2014) distinguish between the process of theorizing a new idea (articulating and justifying why the idea is better than previous ideas) that occurs across organizations within a field, and translation at the local organizational level. This reminds us that innovation does not occur until real organizations do the heavy lifting of adapting and implementing those ideas within their local contexts. Translation

then “links individual organizations to their organizational fields by emphasizing how ideas, which are more or less in fashion, travel into and are institutionalized within specific organizations” (Nielsen et al, 2014; 170).

In addition to the local and situated implementation of material technologies and practices, translation involves a great deal of conversation as organizational members “filter, tailor, and reformulate the discourse.” (ibid). The link between discourse and practice in organizational change is fundamental (Heracleous and Barrett, 2001). Nielsen et al. remark (2014: 171): “[f]ocusing on practice also implies that actors create and transform linguistic as well as material objects as IT ideas travel across an organizational field.” This suggests that discourse is a principal means by which ideas enjoy public mobility in the larger inter-organizational community and get adapted to local conditions by way of internal exchanges. As IT ideas are by their very nature about changing how things get done, the discourse aims at persuasion and the transformation of beliefs. Moreover, the discourse will evolve in substance over time, and its impact on everyday practice can be a matter of degree.

When translating concepts from macro environments into organizational practices, rhetorical devices are used to convince and enroll people into the rationale that will guide action. Translating and diffusing practice requires persistent and significant change in the types of conversations that take place as part of actors’ work practices within organizations. Through translation, a new IT idea that enters an organization from the broader institutional discursive milieu is appropriated, tailored, and eventually absorbed. Yet, there is limited research that shows the links between industry-level ideas and specific organizational actions. A noted exception is the study of ERP software products and their so-called ‘best practice’ design, which is sold internationally and yet fails to work in many local organizational contexts (Wagner, Scott

and Galliers, 2006). Organizations are persuaded by the rhetorical discourse of ERP as a precursor to organization success; the software is then purchased and implemented based on templates that may work theoretically but cost millions of dollars in customizations and adaptations to get the software to work locally (Wagner and Newell 2006) Similarly, Vaast and Walsham (2009) coined the term “trans-situated learning” to explain how distributed teams within a public health administration were able to use a standard information infrastructure to support communication and work practices in a manner that was both local and universal (“local universality”). Pollock and Williams’ (2009; 2011) research on Gartner Inc., an IT research company, showed how Gartner aimed to solidify itself as a thought-leader through the creation of proprietary industry ranking tools and research products. The authors’ examined the industry-level actions taken by Gartner to influence the software ecosystem in terms of product inclusion within a ranking tool (Pollock and Williams 2011). The findings show how, by limiting the number of vendors to include in their Magic Quadrant tool, Gartner caused individual analysts to be more lenient with the software categories in an attempt to work-around the tool’s boundaries (Pollock and Williams, 2009).

Following these studies, we focus on the link between industry-level ideas and specific organizational actions by illustrating how work practices at PL rely on the persistence of a particular ideology expressed in the Agile Manifesto (Figure 1). In doing this, we show how the ideology is not simply translated to support new organizational practices, but rather it has a more continuous role in maintaining these local practices over time. Other researchers have considered the recursive relationship between public ideas and the local practices (local practices in turn influencing the public idea – e.g., Nielsen et al., 2014) (as depicted in Figure 1 side A). There has been limited examination of the role of public ideas in maintaining local practices over the longer-

term. We examine how both employees and managers repeatedly invoked an ideology. This shows that, rather than being absorbed into everyday work practices, the ongoing organizational narrative actively invoked the broader framing and values engendered by the public discourse. We also illustrate how this public ideology conveyed more than simple “lip service” to a grand idea, by showing how work practices, technologies, and discourse helped reinforce the translation and maintenance of such ideas locally (as depicted in Figure 1 side B).

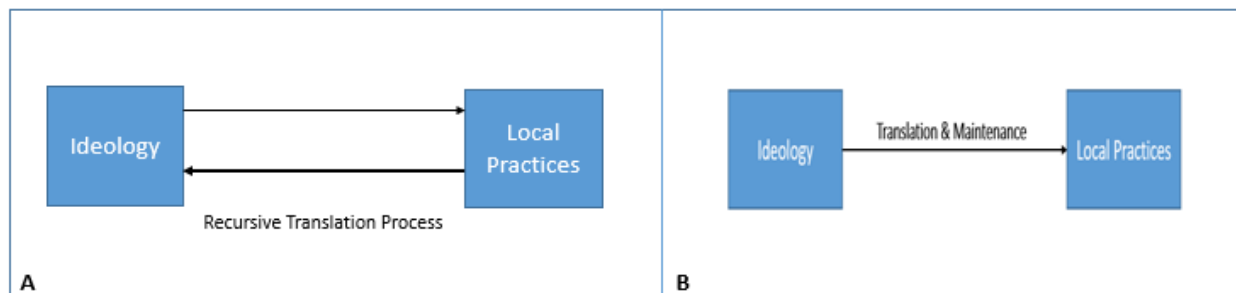


Figure 1. Existing Translation Research vs Study focus

Systems’ implementation literature makes it clear that taking the broad public idea about an IT-related innovation and giving it material reality within a particular organization can be a complex and potentially disruptive undertaking (e.g. Scott and Wagner, 2003). One way to think about this local and situated translation of a new IT idea is to consider how that idea is operationalized through the creation of new work practices and/or the adaptation of existing ones. The material realization of work practices, whether in the form of activities, representational artifacts (such as written procedures), tools and technologies, or the spaces in which work takes place, requires an exploration of how organizational practices are necessarily social and material in their nature.

There are different approaches to studying how the social and material interact, all broadly taking a relational perspective that considers the ‘joint agency of the social and the material, acting together’ (Stein et al., 2014: 157) but differing in terms of ‘how they theorize the status of nonhuman agency relative to human agency’ (Feldman and Orlikowski, 2011: 1244). We discuss this next.

2.2. The Socio-Materiality of Organizational Practices

The relations between the social and material aspects of technology has been a subject of central interest in the information systems discipline over the past several years (Orlikowski, 2007, 2009; Leonardi and Barley, 2008, 2010; Orlikowski and Scott, 2008; Leonardi, 2013). Our discipline thereby joins numerous others in making *the material* a focal concern in the study of the human condition (Hicks and Beaudry 2000), including history, anthropology, sociology, organization theory, philosophy, cultural studies, and studies in art and design (Appadurai 1986; Baudrillard 1995; Brown 2001, 2004; Daston 2004). In information systems this focus on the material means recognizing the performative role of technologies along with “institutions, norms, discourses, and all other phenomena we typically define as social” (Leonardi 2012: 34). Information Systems scholars (Orlikowski, 2007, 2009; Orlikowski and Scott, 2008; Leonardi, 2011; 2012, 2013; Leonardi and Barley, 2008; 2010; Stein et al., 2014) adopt various perspectives on the social and material relationship in order to provide insight into the ways in which humans (social) and technologies (physical and digital artifacts) exist in relation to each other. Leonardi (2013) suggests that they can be broadly characterized as taking one of two philosophical perspectives: critical realism and agential realism.

Agential realism holds that the technologies have no properties, functions, or meaning independent of their uses but play an integral role in defining what it means to be human, both in

a general sense and in relation to specific technologies in practice settings (Orlikowski and Scott, 2008). The critical realism view considers the material and social as empirically separable entities that can be studied to understand how they connect and influence each other in practice (Leonardi, 2012). In this paper we draw on this socio-material perspective because it has been identified as providing a more helpful lens for empirical research to understand the role of the material (and social) in everyday organizational practices (Mutch, 2013). In particular we draw on the work of Leonardi (2011, 2013) and define materiality as the inherent properties of an object that travel across individual contexts of use. Material *agency* describes how the object acts when humans interact with the object. This distinction is important because materiality enables and constrains material *agency* and also *social agency* – what it is possible to do in particular social contexts. For example, digital photos have meta-data that describe attributes about the image such as location, date, time, and image size. These properties follow digital photos across contexts of use. However, these properties do not automatically organize themselves across images by location, until a user directs that materiality to categorize itself as such. In other words, individuals never act alone but are enabled by material artifacts as well as by social norms and conventions (Pollock, Williams and D’Adderio, 2007).

This state of affairs is described as *an equipped context*: work environments are not neutral containers but are prepared with various artifacts and conventions in ways that make it possible to accomplish specific tasks, precisely because the context is “equipped so as to elicit their habitual use” (Gherardi, 2012: 98). In making this observation, Gherardi cites the work of Whalen et al. (2002), who describe operators’ practice in a call center as seemingly choreographed because, when they take customer calls, they appear to have everything ready-to-hand to take notes, and to ask and answer questions. For example, the computer screens the operators use do not include the

customer's name once they have input a question based on the individual's account (materiality). However, the operators have already noted the name on a pad of paper that was kept in "just the right place," which enables them to address the customer throughout the call. The idea of equipped context, then, recognizes how the social and the material influence each other in the accomplishment of everyday practice.

The very substance and meaning of individual action therefore depends on structured interactions with the materiality and material agencies of space, digital and physical technologies, tools, and other people in order to give it shape and direction. Action is thereby always relational, with the individual's power to act afforded and constrained by myriad materialities such that all practice is inherently socio-material.

Beyond this, the relationship between broader public discourses and the materialization of organizational practices is also of interest to scholars (Orlikowski and Scott, 2008; D'Adderio, 2011; Hultin and Mahrng, 2014) but as yet the socio-material lens "has not sufficiently accounted for and linked institutional macro-structures with underlying interpretive processes through which these practices are enacted" (Hultin and Mahrng, 2014: 130). Furthermore, artifacts represent the construction and classification of information according to specific perspectives that are legitimized within the organizations that have created and used them (D'Adderio, 2011). Such artifacts often emerge in situations where different agencies, with different objectives, contend with one another, so that "each agency had to compromise and at least partially accept the other's program, meaning that only portions of their world were realized. The world that ended up existing was thus ... a compromise, a patchwork containing elements from competing worlds" (p. 221).

Theoretical treatments have acknowledged that there is a recursive relationship between specific local practices and specific public ideas, with local practices leading to the adaption of

public ideas over time. But there is little study of how the public idea sustains local practices over the longer term. In preparation for investigating this phenomenon, the following section presents the study's methodology.

3. METHODOLOGY

We conducted a qualitative study (Denzin & Lincoln, 1994) that seeks to understand agile software development in practice, including how the arrangement of spaces, technologies, roles and activities are realized within a firm known for completing all its projects using agile software development.² Because of this interest, we sought access to an organization well known for agile software development in order to conduct an on-the-ground exploratory study with the objective of contributing to the socio-materiality literature. Access to Pivotal Labs was approved and we adopted an ethnographic approach recognized as effective for investigating research issues in context (Schultze, 2000). In addition to the field study, we investigated web-based content that referred to PL practices. Given our interest in understanding how agile methodology is enacted in practice, the study is designed to seek 'validity...not [from] the representativeness of the case in a statistical sense, but on the plausibility and cogency of the logical reasoning used in describing the results and in drawing conclusions from them' (Walsham, 1993, p. 15). Our qualitative design enabled us access to multiple interpretations of the work practices to improve the credibility and potency of our analysis.

3.1. The Context: Agile Software Development

² Pair Programming at Pivotal Labs: PL completes 100 percent of its projects via Paired Programming (PP)
<https://blog.pivotal.io/labs/labs/pair-programming-considered-extremely-beneficial>.
<https://labs.ideo.com/2015/09/22/pivotal-experience/>

Today agile software development is written about in the popular press as if it were the latest reality-television drama. The media relate the controversy between those who claim that agile is dead, those who revere the approach, and those who have profited from its popularity (there are agile coaches, consultancies, a non-profit organization Agile Alliance, trainings, books, conferences, and more).³ Several initial Manifesto signatories founded the Agile Alliance which is now a membership-based organization that, according to Wikipedia “promotes software development according to the [M]anifesto's values and principles”. The Alliance website houses historical documents including a listing of all initial and subsequent Manifesto signatories, and informational content about the Manifesto’s values and accompanying principles.

³ <https://www.theatlantic.com/technology/archive/2017/12/agile-manifesto-a-history/547715/>

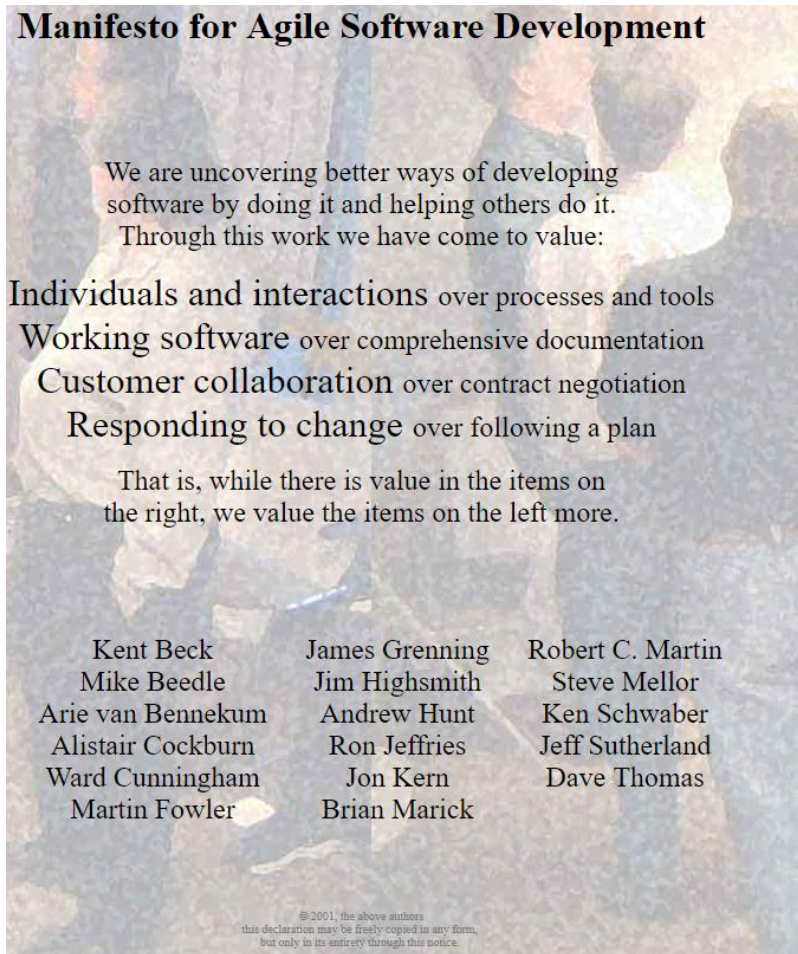


Figure 2: Agile Manifesto as shown on the Agile Alliance web page

The Manifesto articulates a philosophy of software development that outlines four values (Figure 2) and 12 supporting principles for agile development (Table 1). The Manifesto does not provide specific recommendations for enacting these big ideas, but rather provides an anchoring ideology for those seeking to engage in agile software development.

<i>Value 1 [V1]: Individuals and interactions over processes and tools</i>
Principles
Motivated Individuals [V1Pa]: Build projects around motivated individuals. Give them the environment and support they need and trust them to get the job done
Maintain a constant pace [V1Pb]: Agile processes promote sustainable development. The sponsors, developers and users should be able to maintain a constant pace indefinitely

<p>Technical excellence [V1Pc]: Continuous attention to technical excellence and good design enhances agility</p> <p>Self-organizing teams [V1Pd]: The best architectures, requirements, and designs emerge from self-organizing teams</p>
<p><i>Value 2: Working software over comprehensive documentation</i> [V2]</p>
<p>Principles</p> <p>Continuous delivery of valuable software [V2Pa]: Our highest priority is to satisfy the client through early and continuous delivery of valuable software</p> <p>Quick working software delivery [V2Pb]: Deliver working software frequently, from a couple of weeks to a couple of months, with preference to the shorter timescale</p>
<p><i>Value 3: Client collaboration over contract negotiation</i> [V3]</p>
<p>Principles</p> <p>Daily cooperation between users and developers [V3Pa]: Business people and developers must work together daily throughout the project</p> <p>Frequent face-to-face communication [V3Pb]: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation</p>
<p><i>Value 4: Responding to change over following a plan</i> [V4]</p>
<p>Principles</p> <p>Adapt to change [V4Pa]: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly</p> <p>Simplicity: [V4Pb]The art of maximizing the amount of work done is essential</p>

Table 1: Agile Manifesto Values & Principles⁴

3.2. Case Description

The study took place in two locations of Pivotal Labs, a division of Pivotal, a technical consulting company with two-thousand employees focused on web and mobile application development and enterprise automation. Pivotal has offices in North America, Europe and Asia. PL clients have included Google, Twitter, Groupon, Best Buy, and Salesforce. PL also created and maintains a web-based agile project management software called Pivotal Tracker. In April 2018, PL was listed on the NY stock exchange. All client contracts are based upon the agile methodology therefore

⁴ Some of the principles can be linked to more than one value but for the purposes of this study we have grouped them alongside the primary value in order to use this framework in the analysis.

PL requires the client to designate a Product Manager to be a member of the development team and involved in all aspects of the project.

PL offices are open plan and modern. One is likely to see developers organized in small groups occupying a small part of the long rows of tablespots with computers. The open space facilitates frequent rotation of programming pairs and regular communication across team members (Image 1).

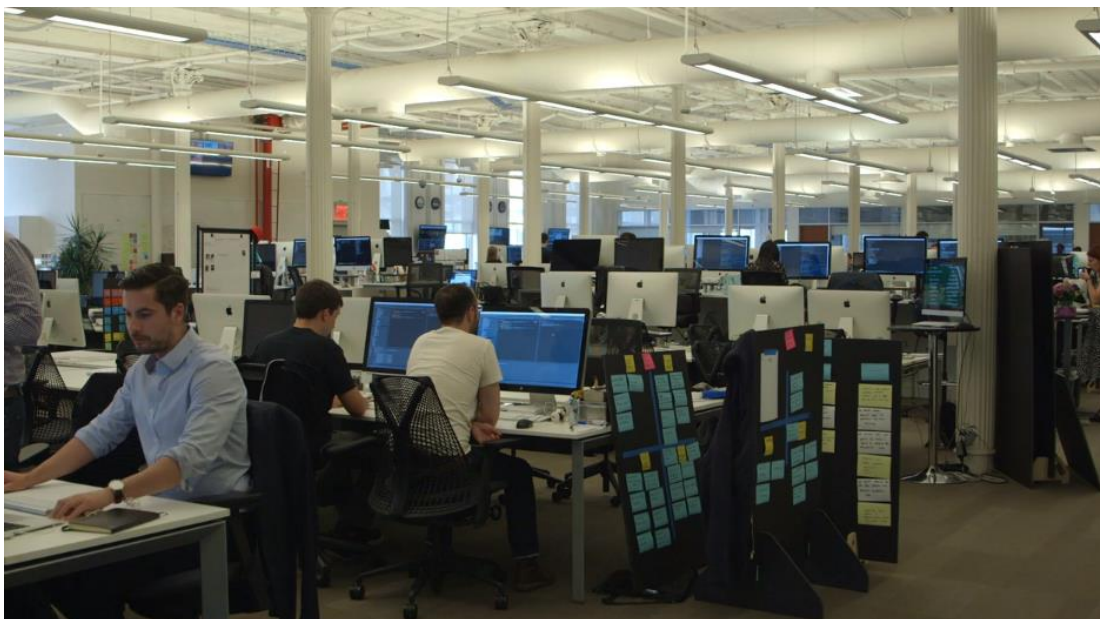


Image 1: Creative Commons <https://vimeo.com/132237986>

PL developers spend the majority of their time working in groups of two pair programming. PP stations exist throughout the open office; each includes one large-screen desktop computer with two sets of human interface devices. This standard configuration is designed to allow each member of the pair to contribute equitably as they trade leadership for writing code by regularly switching back and forth between their keyboards and mice for the duration of the workday (7 to 7.5 hours), using their own keyboard and mouse while having a conversation about the work.

Multiple pairs make up a project team, and pairs typically rotate on a daily basis. This removes constraints at the project level because more individuals with knowledge of the code are available to work on each project and the project benefits from a diversity of views (Figure 3).

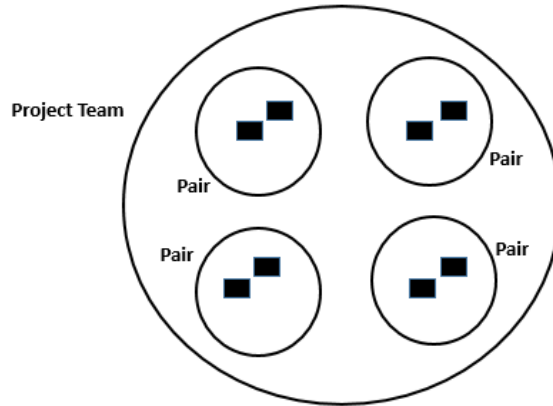


Figure 3. Pairs within Project Team

The Pair Programming Matrix is used by PL developers to ensure systematic rotation, so that each project member is paired with all other project members at one time or the other (see Figure 4 below from a PL blogger writing about the Matrix. This shows, for example: Porky Pig’s column has paired with all team members except Sylvester).

The table is a 10x10 grid representing a Pair Programming Matrix. The top row is a header with cartoon characters: Sylvester, Porky Pig, Daffy Duck, Porky Pig, Sylvester, Sylvester, Sylvester, Sylvester, Sylvester, Sylvester. The first column is a header with cartoon characters: Porky Pig, Porky Pig, Sylvester, Sylvester, Sylvester, Sylvester, Sylvester, Sylvester, Sylvester, Sylvester. The cells contain numerical values representing the number of pairings. A blue arrow points to the cell containing '0' at the intersection of the 3rd row and 4th column. The bottom two rows are labeled 'Solo' and 'Out'.

	Sylvester	Porky Pig	Daffy Duck	Porky Pig	Sylvester	Sylvester	Sylvester	Sylvester	Sylvester	Sylvester
Porky Pig	1									
Porky Pig	1	1								
Sylvester	1	0	0	1						
Sylvester	1	1	2	1						
Sylvester	2	1	1	0	1					
Sylvester	1	0	1	0	1	2				
Sylvester	2	1	1	0	2	2	1			
Solo	0	0	0	0	0	0	0	0	0	Solo
Out	0	0	0	0	0	0	0	0	0	Out

Figure 4. *Pair Programming Matrix Template Example from Pivotal Labs*⁵

Meetings and community gatherings are an important part of PL's work cycle. Short Stand-Ups are held every morning and all employees and client representatives are expected to be present. Longer Retrospectives occur fortnightly on Friday afternoons within each team. Inceptions are formal client initiation meetings where the PL approach to software development is outlined and the client works with a PL team to frame the project. PL also hosts monthly public happy-hour gatherings organized around a particular topic related to agile methodology.

3.3. Data Collection

Our field study at PL was an “ethnography of objects” (Gheradi, 2012), paying particular attention to the material tools and social norms that allowed the software developers in this setting to practice an “improvisational choreography” (Whalen et al., 2002). Our second data collection effort was based on a content analysis of three types of web-based writing. First, we accessed blog posts through the company's website. Second, we included external blog posts accessed via Google searches. Third, we collected all content related to PL on a popular employer review website. These were collected over a two-year period to gain insights into the developers' reflections about their work.

3.4. Exploratory Study

We collected 80-hours of non-participant observational data of software developers, clients, and staff as they worked. In addition, we conducted 28 hours of interviews. The observations took place at two office locations (Boulder, Colorado and San Francisco, California) and occurred during two visits of four days each. Being present at the offices resulted in informal conversations

⁵ <https://content.pivotal.io/blog/pair-programming-matrix> author states one may freely copy.

between the field researcher and the PL employees and clients. Observations were conducted in a non-obtrusive manner and were coordinated with the office manager at the start of each workday. Each observation session lasted the full working day (8 hours) including lunch and a monthly public talk on one evening (additional 2 hours). The field researcher kept a research journal that helped her track these observations and notate points of connection between observations and content from interviews and informal conversations.

Two programming pairs were observed, one at each location, and the researcher took photos of the work as it was occurring. In addition, the researcher asked clarifying questions over lunch and at the end of the workday to ensure her interpretation of what was happening was accurate. In addition, the researcher observed and videotaped a full-day client Inception after receiving permission from the business owner who was engaging PL and was present for the meeting. The researcher's presence during these observations was explained to the employees and clients during the morning Stand-up Meeting.

The videos from the client Inception and Pair Programming observations were transcribed and supplemented with non-verbal content based on the field researcher's notes to capture complex and distributed activities (Ruhleder and Jordan, 1997). Detailed notes were taken during the observation and fleshed out each evening when the researcher returned to her hotel.

The researcher received a detailed tour of each office that included a description of the structural elements found on the programming floor and how these physical artifacts supported the work that was being done at PL. For example, large video monitors were affixed to the walls within view of everyone working in the open-plan office. These monitors provided color-coded abbreviations of ongoing projects and their status. Photos of the physical space were taken and the tour was recorded and transcribed.

In order to collect more focused data, the field researcher interviewed 30 employees and 5 clients. These interviews lasted between 45–60 minutes and were recorded and transcribed verbatim. When citing interview data in our analysis, we list city where the interview was conducted as well as a number showing from which interview the narrative segment derives. We conducted these interviews during lunch periods, after work hours, or via phone at scheduled dates.

3.5. Content Analysis of Web Documents

Given our goal to examine the relationship between broad ideas and organizational work practices, we collected PL website content and saved them in text-based formats for qualitative analysis. We also saved screen shots of the pages to understand layout of text in relation to images. The web pages included customer-facing information about the company as well as the Pivotal blog entries. The blog posts hosted on the Pivotal website were written by current and former employees and were published with the hashtag #agile. In our analysis, these data are identified as blog content along with the date we accessed the post and the blogger’s initials. In addition, we included blog posts not hosted by Pivotal by searching “pivotal” and “agile manifesto” as well as “pivotal” and “pair programming”. These two types of blog posts totaled 67 pages of narrative data (13,314 words) containing 234 passages reflecting on organizational practices that evoked one or more of the four values of the Agile Manifesto, or the nature of PP at the company.

In addition, we collected 116 narrative reviews of PL on a popular employer review site, Glassdoor, that allows individuals with experience of a company to provide details about salary, work conditions, and employment interviews through narrative accounts. According to the terms of the website, the reviews are not able to be modified by employers. These data totaled 57 pages of text (14,800 words). We identify these data by their source as well as the date we accessed the content.

3.6. Data Analysis

We triangulated data from multiple sources totalling 250 pages of text. In order to analyze these data, we followed the iterative process recommended by Corbin and Strauss (1990) and Miles and Huberman (1994). We traveled back and forth between the data and an emerging structure of theoretical ideas that focused on the relationship between public ideology and translation and maintenance of organizational work practices. Given our interest in socio-materiality, we focused on themes in the data related to everyday practices, the “equipped context,” and agility outcomes. We identified themes including “flexibility”, “agile identity”, and “Pivotal culture”. We found that the Agile Manifesto remained an active ideology in the culture of the firm, reinforcing and imparting meaning to actions and identities. For example, both new and a veteran PL programmers referenced the Agile Manifesto in their interviews when describing why Pair Programming was an important daily practice. We also found some negative articulations of PL work practices in the blogs and the Glassdoor data (e.g., “If nothing else you’re all very consistent.”, “No flexibility in schedule especially if you're a developer. Eat breakfast before 9, standup at 9, eat lunch at 12:30, go home at 6.”). A third pattern from the data related to what we call Pivotal Exceptionalism, where the work practices being described were couched as superior to alternatives. Based on these initial patterns, each author independently coded the narratives associated with the four values [V1-4] and associated principles [Pa-d] of the Agile Manifesto as described above in Table 1. For example, referring to the standard working day that is described in the above quote, this would be coded as relating to value of ‘individuals and interactions over processes and tools’ [V1] and the related principle, ‘maintain a constant pace’[Pb].

Passages were coded when the segment reflected practices that explicitly cited Manifesto values or reflected the values. We used the code [**Agile Doctrine**] when the informant was

invoking Agile doctrine in a broad but relatively explicit way—broad, that is, in that it was not specific to any particular Manifesto value. We then identified instances where the *fact* that Pivotal is expressly an Agile development consultancy seemed to be important to the blogger, especially in relation to personal and collective identity. These most often took the form of a kind of expression of *Pivotal exceptionalism*. These were marked with the code **[Agile Identity]**.

We then collected the blog posts as a source of text reflecting on *the role of norms* in reproducing and stabilizing the structure-of-process at Pivotal. Subsequent coding identified passages that spoke clearly to such norms and coded these with the tag **[Norms]**.

4. FINDINGS AND ANALYSIS: PRACTICING AGILE SOFTWARE DEVELOPMENT

The empirical research in this paper aims to understand *how* the Agile Manifesto is reflected in, and sustains the work practices of agile software developers. To this end, we focus on identifying local organizational practices and show how the chosen socio-material arrangement links these practices to the ideology of the Agile Manifesto. Then we demonstrate how this link is sustained through three mechanisms that support the continued relevance of the public idea as a prop for these socio-material practices. This leads to our discussion of a cognitive bridge which depicts the collective role of these mechanisms in the ongoing translation of institutional ideas into organizational practices.

4.1. The Socio-material Translation of Organizational Practice and Their Link to the Agile Manifesto

The Agile Manifesto does not provide specifics about how to enact its principles, so the question of how to “do” agile development remains open to interpretation. In this sub-section, we analyze several socio-material practices that typify software development work at PL and are manifestations of particular Agile Manifesto values and accompanying principles (Table 1). By

doing this we show that the reproduction of local practices rely upon the ideology of the Agile Manifesto. We take each value in turn, but recognize that the practices being described often embody more than one part of the Manifesto's stated principles. These principles are summarized alongside examples of socio-material practices at PL at the end of this section (Table 2).

Value 1: Individuals and interactions over processes and tools

The first value cited in the Agile Manifesto emphasizes individuals and interactions over processes and tools (V1), this is evidenced at PL by the specific configuration of each office (Image 1). All PL offices provide a standard workspace configuration. No laptops are allowed on the floor, and there is no email at the PP stations. Email kiosks are available but are elevated off the ground, and so require standing. The field researcher never observed any PL employee checking email on the programming floor and private calls and meetings take place away from the main workspace. However, throughout the day team members get up for short breaks—some might wheel around on scooters or play a game of “Pivot Pong”. PL offices have fully-stocked kitchens and an office manager who is called “mom” and is responsible for providing everything employees need in-house.

The lack of machine ownership can be interpreted to align with the first value's emphasis on low priority given to tools. Pivotal's full kitchen and office mom are attempts to give employees the environment and support they need (V1Pa) so that they need not leave the premises during the work day. PL employees agree to the material arrangements that are preconfigured for them by the company and this material arrangement limits employees conceptions of what changes are possible because they are “constricted to a large degree by the initial materiality that they encounter” (Leonardi, 2013: 69).

The structural environment at PL is the backdrop against which the motivation (V1Pa), consistent pace of work (V1Pb), technical excellence (V1Pc) and self-organizing teams (V1Pd) exist. This is evident in the blog posting by JP, a PL employee:

If left to my own devices, I will often find a dozen other things to occupy my time, rather than writing code, even if I find the code interesting. This is, in my opinion, one of the biggest advantages of Pair Programming. Your pair is simply not interested in your email. Because of this, a pair of developers is able to stay on task all day long [V1Pa; V1Pb]. The productivity boost from this is tremendous. I simply had no idea how productive I could be until I started Pair Programming. There are many other benefits to pairing: training of the weaker developer [V1Pc, d], knowledge transfer, shared ownership of code [V1Pc, d], better code design [V1Pc]. If I could only pick one benefit though, keeping everybody on task would probably be it [V1Pa]. (4/26/13).

At PL, Pair Programming is the central practice for realizing agile principles and, as such, makes up the bulk of developers' work. PL developers talk about how PP increases the number of people on the team who understand the project thereby decreasing the likelihood of a project failing if one person were to leave (referred to as "bus count"), as illustrated in a PL developer's explanation of how PP works:

On our current project, we started with a new technology that only two people in the project had had experience with out of six developers ...The two people worked together for a little while on it, built a base and now I think everybody is on the exact same plane with working with this tool and technology because you're pairing every day and you have to share knowledge. ... Because we rotate pairs (V1Pd) and because we're constantly talking

through problems and solving them together, even if someone is like the expert at this technology that we adopted in the beginning, by the end, there's no expert ... You learn a lot more (V1Pc) and you're never slowed down because you can't find that one guy who remembered how to do that one thing. We all end up on the same plain. (Interview #2 Boulder office).

The daily rotation of pairs means individuals with knowledge of the code are more numerous and also means that team members develop a holistic view of both the code and the project goals (V1Pd).

At PL, the structure of Pair Programming affects the work. In the below quote, a PL developer working in the San Francisco office elaborates on how structural elements of PP translates to standardized patterns of work:

We swap people around and expect them to be adding value on a new project day one. We do that by standardizing a few things about our workstations. For starters, every computer is basically the same minus the project code itself ... The choices may not always be “the best” when getting down to specifics, but it is one less hurdle to jump when you find yourself on a new project with new faces, new clients, and new domain—at least your computer feels like the home you just left. It also means I don't have to think about how to set up a machine when starting a new project, the collective knowledge of Pivotal workstation config is captured in our machine image. (Interview #2 San Francisco office).

This blog segment demonstrates a material standardization that accompanies—and helps to reinforce—the behavioral standardization promoted by norms. The ways that materiality tends to norm behavior is also shown below through a blog segment by a PL employee related to the Pivotal Tracker:

The real magic of Pivotal Labs that you might not notice at first is Pivotal Tracker. It may look like an ordinary project tracking system, but it does two things very well that most software fails to do: 1) Tracker forces clients to prioritize the work we do for them, and 2) it serves as a central communication piece. (WR, 4/26/2012).

The representational artifacts in this case illustrate how the material can play a direct and integral role in bringing larger institutional thinking into the everyday practices of the organization.

Value 2: Working software over comprehensive documentation

Valuing working software over comprehensive documentation is the second item called out in the Agile Manifesto. The socio-material practice of consistently delivering valuable software (V2Pa) that works (V2Pb) and is measured often (V2c) relies heavily on the test-driven development approach and the tracking of a team's velocity.

When a pair sits down to begin programming, their first step is to open the Pivotal Tracker application where the tasks are organized into three categories of work: bugs, chores, and stories. The latter are small chunks of functionality written as narrative descriptions of what needs to be accomplished, rather than the method for how to accomplish the tasks. These stories are created and ranked in priority order with the client during the initial Inception meeting. The team estimates how much time each story will take to complete. At the end of a 40-hour work week, the team compares their estimated productivity against actual work completed to come up with a project velocity. The velocity enables PL to provide clients with reliable time and functionality estimates. The goal for each team is to maintain that velocity over time, rather than cram work in as the deadline approaches and the client's scope of system functionality increases.

In order to maintain this velocity, pairs must successfully test each piece of functionality that is being written in software code. This is described the following segment from a 2013 blog post by a PL developer:

Since all of our code is test driven, you are focused only on making a single test pass at a time [V2Pa]. You don't have to worry about the whole system. Once your test passes, you can run all of the tests, and they will tell you if you've broken any other part of the system [V2Pa]. Test driving code also tends to make code that is more loosely coupled, meaning it is easier to change one part of the system without changing the others [V2Pa, b]. (JP 1/20/13).

Each time a test is created and run, the developers also do regression testing to ensure that the current test does not introduce problems into the preexisting code. Pairs then write code that will pass the test. When the test passes, the code will turn green in the test environment. This approach works for all levels of testing and enables the customer to view the feature and provide necessary feedback before the code is officially checked into the code repository. Furthermore, the status of a team's code (red, green) is reflected on the large video monitors hanging from the office walls (V2Pb).

After groups of code pass, this functionality is shared with the client who provides feedback and sets future development priorities (V2Pa). When the client is regularly viewing functionality, it helps the PL team to shift the focus from documenting what they are doing to creating chunks of working software. We see in this testing environment that the material and social arrangements are working in concert to create the *power* for developers to act. As Pollock and D'Adderio (2012) note, individuals never act alone but are rather "propped up and aided" by all kinds of material artifacts as well as by social norms and conventions. Indeed, the very substance and meaning of

individual action is relational (Orlikowski, 2007), and depends on structured interactions with space, materials, tools, and people to give it shape and direction.

Value 3: Client collaboration over contract negotiation

This is similarly the case with the Agile Manifesto's third value, which states that client collaboration is valued over contract negotiation (V3). As stated in a company blog post "Our clients see the results in the way which we interact with them to deliver exactly what they need" (WR 4/26/12). An integral part of many PL projects is the education of the client team, so while the cost of hiring PL is far higher than many other IT services firms, PL argues that this is offset by training clients to be agile once the engagement is concluded. Thus, the way they consider software engineering is fundamentally changed, from one of Big-Design-Up-Front to iterative client-lead development. The practice of pairing with clients allows for continuous recommitment to the ideology by both the client and the PL developer.

There are few stronger signals of commitment to a value in business than turning away clients who do not respect those values. The fact that PL has turned away business based upon a lack of commitment to client collaboration (value 3), ostensibly signals that the Agile Manifesto is alive and well at PL. PL requires that each client assign a responsible Product Owner who pairs with team members (V3Pa), engages in frequent face-to-face communication (video or in-person) during team meetings (V3Pb), and approves the functionality (story) as it is sent to them in the Pivotal Tracker. This process is described in an interview with a PL developer:

Tracker points out [assigns points] that this is our expectation of the next iteration. So it really forces the Product Owner to make decisions about what's most important to the business. It's not an exact science but if the Product Owner sees a three-point story in a

current iteration and says “well, that’s nice but we have this big conference in two weeks. So I want to move up this other story that does our marketing pages and I can know the effect and I can constantly look at the projection and I can make decisions on what’s the focus of this 40-hour week and how does that predict the future”, instead of saying “oh, we have to get all of this done” (interview #4 Denver office).

The materiality of Tracker is entangled with the social world of PL software development and vice versa. This helps to reinforce the actions and identities of agile software developers at PL.

Value 4: Responding to change over following a plan

The fourth value stated in the Agile Manifesto—responding to change over following a plan (V4)—is also evidenced through the design of the Pivotal Tracker which divides programming tasks into three buckets: prioritized stories, then bugs, and finally chores. Categorizing work in this way makes sure the velocity remains consistent and non-essential work does not consume the majority of a day. This way of working is illustrated with the following blog segment from a PL employee:

This newsroom style is the key to success within a business that is constantly buffeted by a changing environment, leaping to seize new opportunities, and reacting to the actions of competitors. Like journalists, our objective must be to deliver each new story—each requested systems change—with speed and craftsmanship. This is what we mean when we say that we strive to be agile (V4Pa). (JH 2/9/14).

The art of maximizing the amount of work not done (V4Pb) is a principle that requires a deep understanding of the work that is to be done. This point is supported by the regular client review of stories in Tracker, and their ability to redirect the development work. These regular check points

mean that the amount of unnecessary work is limited to the time since the last check-in (usually a week).

Retrospective meetings that occur every two weeks provide structured time for the project team (including the Product Owner) to reflect on the work since the last meeting of this type. These meetings are in addition to using the Tracker to monitor work. During these meetings the white board, and sticky notes are used to list action items and proposed behavior change (V4Pa). Retrospectives follow a standard convention. On the whiteboard, a smiley face is used to show what went well, a sad face is drawn to show what the team can improve on, and a neutral face indicates what questions still exist. This helps resolve issues and prepare activities for the next iteration of work.

This section has presented evidence showing the connection between the actors’ descriptions of their work at PL and the external ideas, values, and ideologies as expressed in the Agile Manifesto. The representational artifacts in this case illustrate how the material can play a direct and integral role in bringing larger institutional thinking into the everyday practices of the organization. The maintenance of local socio-material practices at PL are supported by the public idea (the ideology) over time. Next, we consider the mechanisms that support the continued relevance of the public idea as props for these socio-material practices.

Agile Manifesto Values & Principles	Representative Socio-material Practice at Pivotal Labs
<i>Value 1: Individuals and interactions over processes and tools.</i>	
V1Pa: Motivated individuals: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.	Developers are motivated to become the best they can be in terms of technical excellence and knowledge. The PL office environment supports physical and social needs. The team supports their professional development and sense of community. The team is empowered to be successful.

V1Pb: Maintain a constant pace: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.	Standard 40-hour work week with set working hours, pairing 95% of the work day The lack of overtime and deadline crunches enable sustainability of work Pivotal Tracker's velocity calculation supports the management of a constant pace of work
V1Pc: Technical excellence: Continuous attention to technical excellence and good design enhances agility.	Rotation of pairs and Pairing in general helps to share knowledge and increase technical skills across the Pivotal employees. Rotation of pairs and movement across teams enhances agility. Test Drive Development ensures that code is being written that has passed testing – teams do not move forward until they change a red-status project to green.
V1Pd: Self-organizing teams: The best architectures, requirements, and designs emerge from self-organizing teams.	While PL does assign employees to teams, the teams themselves are responsible for rotating pairs.
<i>Value 2: Working software over comprehensive documentation</i>	
V2Pa Continuous delivery of valuable software: Our highest priority is to satisfy the client through early and continuous delivery of valuable software.	Weekly delivery of stories to clients. Clients must approve or send back. Approval of the stories The client directs the functionality in ways they see as valuable and can do this on a weekly basis from within the Pivotal Tracker. Test Driven Development focuses on creating functioning segments of code regularly.
V2Pb Quick working software delivery: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.	Clients are provided segments of functionality on a regular basis and based on their use of the functionality, they direct the next phase of work.
<i>Value 3: Client collaboration over contract negotiation</i>	
V3Pa Daily cooperation between users and developers: Business people and developers must work together daily throughout the project.	Each project has a dedicated Product Owner that represents the client. This individuals works at the Pivotal Offices every week and pairs with PL developers. The client is approving the software functionality and/or redirecting the programming efforts.
V3Pb Frequent face-to-face communication: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation	Since the Product Owner is on-site at Pivotal, face-to-face meetings are supported. Video conferencing is also conducted. The client inception meeting must take place on-site at Pivotal. Pairing, weekly meetings, daily standups, and daily team standups are face-to-face.
<i>Value 4: Responding to change over following a plan</i>	
V4Pa Adapt to change: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly	Retrospective meetings that occur every two weeks are an opportunity for the project team (including the Product Owner) to reflect on the work since the last meeting of this type. The white board is used and action items for behavior change are noted.
V4Pb: Simplicity: the art of maximizing the amount of work not done--is essential	Pivotal Tracker divides tasks into three buckets and stories are prioritized, then bugs, and finally chores. Categorizing work in this way makes sure the velocity remains consistent and non-essential work does not consume the majority of a day. Checking in regularly with the client about the direction of the development limits that amount of code that is written that may later be determined useless.

Table 2: Agile Manifesto & Examples of Socio-Material Practices at Pivotal Labs

4.2. Mechanisms that support practices by connecting the local practices with the wider ideology.

We have shown the ways in which the translation of broad public ideas were locally reflected in PL's socio-material practices. In this sub-section we identify three translation mechanisms that support local work practices by connecting them to the broader public ideas upon which the practices are based: doctrine, identity, and norms. Together these mechanisms create a *collective cognitive bridge* across which local practices join with ideology. This bridge helps to sustain the practices over time.

4.2.1. References to an Agile Doctrine

In addition to the four specific values, the authors found a pattern in the data that referenced agile doctrine—statements that refer more broadly to the larger meaning of the Agile Manifesto. For example, the Pivotal Labs web page features Agile Methodology as the way in which software is built, and reference is made to the Manifesto being enacted in practice through “real collaboration” (Figure 5).



Figure 5: *Pivotal Labs website copy making explicit reference to the Agile Manifesto*

The co-founder’s involvement with agile development since the methodology’s beginnings is evidenced on the website by linking Pivotal Labs to one of the Manifesto’s original creators and founder a version of agile methods called extreme programming (Beck, 2010). Further, the PL web content states the co-founder was “instrumental” in the adoption of agile methods at “some of the world’s largest e-commerce companies”. This shows the original PL business model is founded on agile ideology.

Linking agile methodology to Pivotal Labs is a mechanism that tells the audience (prospective clients and employees) that Pivotal Labs equals Agile. This is further demonstrated in a blog post from a PL employee:

Being an agile shop means we’re always gathering feedback and looking for ways to do our job better. Each project is different, so we have regular Retrospectives to help us surface and adapt to those differences ... I strongly believe that, if left alone, the right people with Retrospectives would arrive at a process that looks a lot like Pivotal Labs does

today—in fact I would argue that’s what happened when the idea of agile was first put to paper by Beck, Fowler and the rest of the agile work group. (WR 4/26/12).

This blog segment, while written by an individual, shows Pivotal Lab’s position on the way to conduct agile development work. It also demonstrates the indoctrination of the blogger who has become an advocate of *the Pivotal way*.

The image of programmers working in pairs is often used by Pivotal when representing the company (Figure 6). This is doctrine on display. The PL website shows four pairs of people sitting at computers either looking at each other or a screen. The image is accompanied by text that, together, demonstrate at least two of the four Agile Manifesto values: V1 – Individuals and Interactions Over Processes and Tools and V3 – Customer Collaboration Over Contract Negotiation:

We are agile to the core and lean in practice. Innovative ideas excite us. Cutting edge technology fuel us. We break down the silos of knowledge and embed members of your team within our own to create a high-energy, fully transparent partnership. (Pivotal Labs landing page as of 8/2018)

Evidence of V1 can be seen by the prominence of people over tools in the image. Evidence for V3 is seen in the both photo of collaboration and the text copy. This image shows pairs working together and the text reads, “We ... embed members of your team within our own to create a high-energy, fully transparent partnership”.

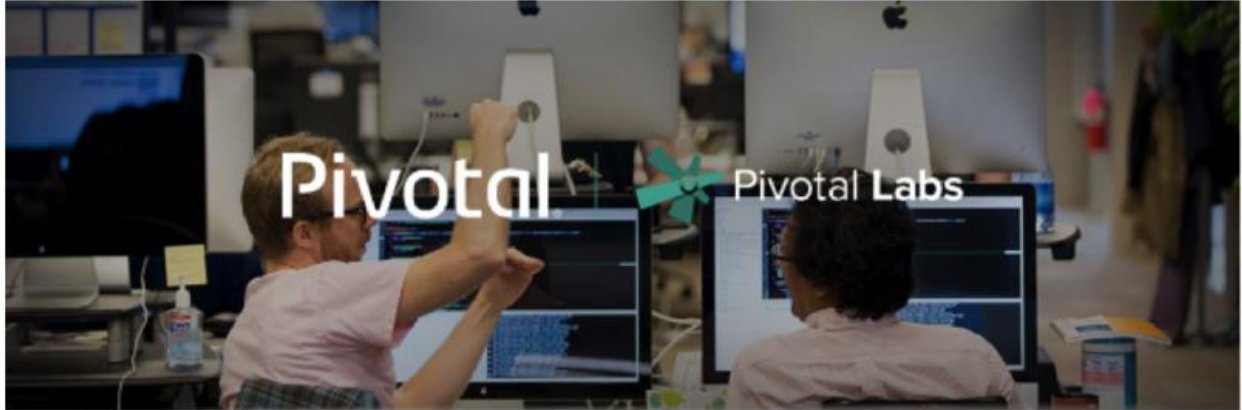


Figure 6: Pair Programming at Pivotal Labs (By Ryanjones10 [CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0>)], from Wikimedia Commons)

The text underscores the value of working software over comprehensive documentation (V2) by referring to Lean Manufacturing in the statement, “We are ... lean in practice” where non-value adding activities are reduced in favor of incremental product improvements. The value responding to change (V4) is inferred in statements such as “We are agile to the core,” “Innovative ideas excite us,” and “Cutting-edge technologies fuel us.” The company website purposely connects PL to the agile ideology—a big idea—of which the company is a part. In this way, marketing and promotional materials are used to make a cognitive association for viewers between Pivotal Labs and agile.

The July 1, 2013 blog post by PL employee JB entitled “Post Your Principles” is an explicit appeal to the ideology that has been implemented in organizational practices and speaks specifically about how to implement the doctrine. The text follows:

Principles Should be Public

True Story

Once upon a time, there was a client who struggled with the Agile process. It's not that he wasn't smart and curious, and it's not that he didn't want to build a great product in a domain he knew lots about, but he had a thorny problem: because he was successful, and because he was smart, he just wanted to build this new digital product the way he'd successfully built (non-digital) products before: top-down and waterfall. He didn't want to make decisions about the details. He refused to look at Pivotal Tracker.

He came to us because he'd heard "Pivotal's the best", but he didn't put together that the process—pairing, testing, short feedback loops, collective ownership of the product, writing the smallest possible stories that deliver user-facing value—were the reason we were successful. We tried to educate, we tried to make the case, we tried to explain Agile, but nothing really worked. He didn't have the patience, and he just wanted to know why we weren't building a product that was perfect.

Everything Changed

One day, a Pivot on the project printed up the Agile Manifesto and hung it on the wall. That's it. Nothing else.

Nothing happened the first day.

Nothing happened the second day.

On the third day, the client, of his own volition, pointed to the manifesto and asked a question: "What's that?". We started to tell him, and that opened up the conversation. He was smart and inquisitive and curious, and the more he learned about Agile, the more he understood why we insisted on tight feedback loops, small stories, and user feedback.

The moral of the story: post your principles. Make them public. Let everyone see them. Make them a part of the space you inhabit, and it becomes a little easier to make sure they're present in your practice.

Update (2014-07-28): In the spirit of *post your principles*, here're a bunch of desktop backgrounds to help keep agile and extreme-programming principles in-view and top-of-mind. Download the full set of Agile Desktop Backgrounds.

The post hyperlinks to the Agile Manifesto. As the text illustrates, the blogger tells the story of converting a skeptical client by simply hanging the Manifesto on an office wall without any comment. The fact that the Manifesto was explicitly visible provided an opportunity to discuss the values and principles and this discussion, in turn, convinced the skeptical client about why the PL ways of working were useful. This explicit link to the ideology both on the web and in the PL office is evidence of the doctrine's direct influence on work practices. Further, the blogger provides a link to a zip file of desktop backgrounds/wallpaper of each manifesto value. This allows developers and clients to keep the Agile Manifesto front-of-mind. As the blogger says, "Post your principles. Make them public. Let everyone see them. Make them a part of the space you inhabit." The artifact of the printed Manifesto itself was the bridge into agile methodology for the client. The Manifesto is invoked to gain and educate new/potential clients and underscore the motivations for methods employed during the development project. Employees leverage the Manifesto, as a type of social movement, to legitimize PL's approach to software development.

4.2.2. Cultivating an Agile Identity

Agile identity can be seen as a kind of "Pivotal exceptionalism" in which employees differentiate themselves and the company from competition. In large part this is because of their intentional association with the agile doctrine, as described by PL executive Edward Hieatt "We have succeeded in creating something unique ... We've challenged conventional methods of software development" (1/7/2013). These expressions of identity are just as important as work practices being linked to the Manifesto because they act as mechanisms that help maintain a bridge between the local and broader public ideas.

The following series of quotes from a single blog post by a PL employee are representative of the perspective shared by employees during the case study. We interpret the sustained

ideological rhetoric as creating something akin to ‘group think’ a collective cognition that links the institutional and the local:

First off, I’m not excited to be called a cult member—the term is derogatory and carries a lot of baggage with it. But if we look at this definition from Wikipedia, the word cult is a “term for a new religious movement or other group whose beliefs or practices are considered abnormal or bizarre by the larger society.” While we’re not “religious,” I would say we’re a “movement” to change the way software is made, and our practices intentionally push the boundaries of what most organizations are comfortable with. But what makes a cult is beyond practices. (LS 4/26/13).

Radicalism as well as discontinuity is being invoked here and, as with the next segment, speaks to the collective cohesiveness of the identity around a highly coherent set of practices. It also speaks to the salience of the agile principles that provide an ideology upon which all practices are based.

I said I thought I knew what people were getting at, but I didn’t fully understand it until recently when I was sitting down with a candidate who was on the edge about accepting a job offer at Pivotal. She had been through our usual rounds of interviews and had some really good questions. I did my best to answer them honestly. At the end, she thanked me and offered, “If nothing else you’re all very consistent.” I indicated I wanted to know more and she shared that she had also talked to Pivots at Meetups and even overheard a couple of us on the train and apparently we “all say the same things about [our] company.” She had to leave, so we said our goodbyes, but what she had said moments before was still washing over me. (LS 4/26/13).

These blog segments show that employees are missionaries. They “drank the Kool-Aid” and now want to share it. Their explicit awareness of the ideological values is evident in the everyday

performance of work as well as how the public narrative of the Agile Manifesto frames the official accounts of PL practices.

Evangelizing PL's approach to agile software development on the company blog is one way that PL shapes the conversation about the company. Detractors of PL also highlight the fact that practices emphasize the translation of big ideas into local practices and they reference the impacts of maintaining those practices over time:

For me, the full-time pairing and fixed hours got to be a serious drag, even though I think it's a huge strength for the company itself and the employees who can deal with it. I'm an introvert, and after a couple of months I just couldn't take it anymore. However, that's just how Pivotal Labs does it—I don't hold it against them. So, in the end not for me, but it might be for you. Due to the nature of Pivotal's practices, I ended up feeling that I truly was merely a cog in a giant machine (or a "Pivot" as the company terms it). Nothing wrong with that for many people, but not my bag after experiencing it for a while. (Glassdoor review from former PL employee, 2/20/2015).

Ninety percent of former employee reviews of Pivotal Labs featured Pairing as a key feature of work in the company. Across all data sources, Pair Programming in teams was the most frequently referred to work practice.

Ongoing conversation between PL employees are necessary for translating concepts from institutional environments. In so doing the discourse is also localized (Neilson et al 2014). In the group interview segment below, we see PL employees engaged in that ongoing conversation about work practices and what they mean:

Employee 1: “Arguing on solutions, we do that always. That’s healthy. We need to do it. Because if we don’t do it, then we’re just going to blindly accept someone’s solution without even questioning like whether it is a good solution or not. You don’t do a great product doing that.”

Employee 2: “Pairing is pretty intimate after you’ve done it for a while. All these guys...I expect you guys to give honest feedback.”

Researcher: “So does it foster a competition?”

Employee 1: “No.”

Employee 3: “No, it doesn’t.”

Employee 1: “It fosters the high five. Like it’s so gratifying to see someone else just crush the problem.

Employee 2: “That’s true.”

Employee 3: “Oh, I think I’ll do this and then you go like ‘oh, my God, that’s so cool. Let’s do it that way’. And then you jump on the train...”

Employee 2: “It’s like that.”

Employee 3: “And often like five minutes later, you go ‘wow, that’s awesome. How about we change this?’...”

Employee 1: “That’s why we work 40 hour weeks because ... we sit down together [to pair] and we stand up together [for Standup Meetings at the start of our day]. And we’re flexible around the edges but the main benefit of the whole exercise comes from doing it together the bulk of the time ... 90 percent, 95 percent is sitting down together.” (Group interview #1 Boulder).

When Employee 1 explains the value of all employees following the same work schedule, we understand that he believes in the practice and associates it with Pivotal exceptionalism. This socialization process is where we most clearly identified norms that support the socio-material practices by connecting the local practices with the wider ideology.

4.2.3. Norms of Behavior

Notwithstanding the casual style of the work environment, the scooters and ping pong, or the intimacy that some get from PP, the evidence points to a need for a more formal, sustained translation between the ideology and the socio-material structuring of work. Pivotal Labs work requires specific structural conditions and behavioral expectations and everyone adheres to certain social practices and uses artifacts in very specific ways. Recall that PP stations have no email or personal software functionality. Phone calls are made away from the programming area. Even time itself is tightly bounded, with office and team “stand-ups” to start the day’s work, pairing to help enforce a constant velocity, and norms that proscribe no work on weekends, evenings, or beyond forty hours per week:

So another important part of Agile is try to keep it to ideally a 40-hour week schedule and not—I mean every project goes in times where you have to do a little bit more but ... most of the time we come in whenever our stand-up time is and ... nine hours later, we leave ... We also kind of advocate an hour out for lunch and a lot of that is not just so we’re here longer, it’s just that when you’re pairing, it’s a little bit more intense. So that break to kind of clear out the thought really helps the quality of the afternoons. (interview #13 Boulder office).

The hard and fast norm of Pair Programming makes up ninety-five percent of the developers' workday. This convention is widely understood as part of Pivotal's culture. This is demonstrated with the following representative quote:

As a software developer, you bring money to the business, so you are treated like a profit center rather than a cost center. You will never work more than 8 hours a day, ever, and if you do, you might be chastised for it [V1Pc]. You will be surrounded by smart people who want to do the right thing [V1Pa]. You learn at an accelerated rate due to Pair Programming every day [V1Pc, d]. If you don't like Pair Programming, you will not like working at Pivotal Labs. If you don't like writing tests before you write software, you also will not enjoy your time there. (Glassdoor review from PL employee, 4/1/2014).

Pivotal employees come to rely on these social conventions:

Since Pivotal developers are so productive during the workday, couldn't we be extra productive by going home and working late into the night? Maybe, but probably not. Working late tends to throw schedules off. It's hard to get to work at 9 if you go to bed at 4. Everybody starting at the same time means everybody pairs. It's one of the few drawbacks of working at Pivotal, your work schedule is not flexible if you are pairing. (interview #10 San Francisco office).

While the following blog segment from PL employee JP does not explicitly refer to norms, it is about the common value that underlies consistent norms:

This constant sharing of ideas and perspectives quickly creates a value alignment without anyone ever having to say what those values are. We even pair when interviewing so we're selecting for those values from day zero. And since we have this strong alignment, it's easier to identify Pivots as a group, or a "cult" if you must, rather than as individuals who

work for a faceless brand. If you want your company values to be represented at all levels, you have to talk about and challenge those values every day so that there is shared understanding, and pairing is a great way to do just that. (1/20/13).

Another interviewee likened the team dynamics to being subject to military-grade discipline and comradery:

It's better to be part of a team. And this goes back to like the military and why a lot of veterans can't adjust to normal life anymore after being part of a tightly knit team in Afghanistan, in Iraq ... That's part of what people get at Pivotal, it's like they become part of a pack and those packs form very quickly through pairing and some of the other kind of communication pieces that we have in place, where the feeling of being in a pack and having success and facing and meeting challenges as a team, getting through them and then succeeding, that's actually much better—that's more rewarding than that feeling of "I'm an individual with specialized skills and people need me". It's more powerful than that. (interview #4 Boulder office).

While intense and exhausting, the "feeling of being in a pack" is more rewarding than being an "individual with specialized skill."

Clients also undergo an intense socialization process when beginning to work with Pivotal Labs. Clients must attend a two-day Inception meeting on site which involves the Pivotal Labs development team and clients in a large conference room with white boards and writable glass walls, video conferencing capability as well as screen projection. In other software development methodologies, the Inception would be considered "requirements gathering". However, the goal of the Inception is "to air all the ideas and assumptions about every aspect of the project so that the entire team starts off on the same page" (Official PL documentation). The clients are expected

to be present for the entire Inception, and to avoid using mobile technology and be fully engaged in the work going on in the room. After observing a client Inception, the first author was told that “this is when clients are enculturated into the Pivotal way of developing software”. Lunch is brought into the room and a few short breaks of approximately ten minutes exist during the eight-hour session. All participants are discouraged from leaving the boardroom for more than a few moments.

The wall dedicated to a white board is used quite a lot during Inceptions, not just for writing but also for affixing colored index cards that list goals, risks, users, and product features. Prior to taping the cards to the wall, all members of the Inception are asked to place three chocolate kisses on the cards that are most important to them. This process helps rank product features and begins the articulation of concrete steps that will be imported into the Pivotal Tracker as stories. The agreed-upon list of stories becomes the first iteration of product functionality delivered by the team. Programming begins directly following the two-day Inception. Part of the Inception involves emphasizing the benefits of programming in pairs and this rhetoric is an important mechanism that links Pivotal back to larger ideas about the value of agile methods. It is worth reflecting here that the principal proposition of PL is that a single developer *cannot* produce what two can, and that PP is not merely about efficiency. The argument made by PL employees is that the efficiency of PP saves time in such a way that it is not necessarily the case that a client pays twice as much for the same result. Two programmers might do the work in half the time, resulting in a levelling of remuneration.

In summary, our socio-material analysis shows the importance of public ideology in translating and maintaining local work practices. Our findings reveal the mechanisms and processes of translating concepts from the institutional environment into local practice. At PL, the

Agile Manifesto was embodied in the performative work practices and experiences including the following: the commitment to human interaction (V1), the dynamic nature of the software production (working software over comprehensive documentation, V2), the strict commitment to client collaboration (V3) and the emergent character of how work gets done (responding to change, V4). We showed how the PP practice (and the entire PL environment, more generally) is a consistent socio-material expression/manifestation of the underlying ideology. The narrative elements in the contributions of informants help to illustrate PL employees' explicit awareness of the ideological principles at work in the everyday performance of work practices. We discuss these findings next.

5. DISCUSSION: THE COLLECTIVE COGNITIVE BRIDGE

We have illustrated three mechanisms that support the local socio-material practices of PL by connecting those local practices with the wider agile ideology. The translation of ideas into local practice relies upon the adaptation and implementation of social conventions and material objects *as well as* actors' conversations about the work (Neilson et al., 2014). Ideas are mobilized inter-organizationally through discourse that seeks to sway and inform beliefs. We see here that, even in the distaste for PL practices, the story remains that PL is dedicated to specific practices derived from the Agile Manifesto's ideology. PL's instantiation of agile methodology as expressed in the Manifesto is assumed to do much of what it promises to do (i.e., increase efficiency, train clients, immerse employees in projects, etc.).

PL employees' work is subject to public illumination through devices (e.g., turning code green to pass a test and video monitors display project status) that constitute a kind of collective panopticon (Foucault, 1995). These social and material arrangements could be interpreted as

limiting the individual developer's freedom, for example, by disallowing flexible working schedules based on personal temporal rhythms (the late night coder, the crunch time sprinter). Indeed, there is evidence that some find these practices constraining and cult-like. On the other hand, our PL informants expressed the opinion that they experienced social conventions and the arrangement of work as liberating. The question of whether this approach is of sufficient value if it results in, for example, burned out employees (e.g., the alienation of the introverted interviewee) and other harms is not without merit. Why is the Kool-Aid any good? However, this is not the primary focus of this particular study. Rather, our focus is to understand how ideas circulating inter-organizationally play a part in the instantiation and maintenance of local work practices.

Our results indicate that what helps instantiate an agile process, is made from the very structure that control-based perspectives commonly discuss as constraint. This structure also allows for the kind of adaptability that molds the code to client needs. In this sense, the programmer, the task, and the socio-material work environment are strengthened through what we define as a *collective cognitive bridge* back to ideology – in this case the Agile Manifesto. As we have described, the bridge to ideology is based on constant referencing back to the Agile doctrine, the acceptance of a 'cult-like' identity in relation to being evangelical about the agile values and principles and the acceptance of norms about how work is done that are strictly adhered to. This cognitive bridge supports the continued relevance of the public idea as a prop for socio-material work practices. These three mechanisms linked local agile methods to larger ideas about “what agile is” in a more abstract and generalized form. In this respect, it is helpful to remember that the Agile Manifesto says nothing about methods for realizing its values. The early explicit adoption of programming in pairs, for example, is the product of local discourse that translates a publicly available new IT idea into terms of the organization's goals, needs, and implementation conditions.

We have seen in our exploration of Pairing that work practices are maintained and renewed through sustained, explicit, and deliberate appeal to the Agile Manifesto and the distinctly public ideology it represents. The empirical insights support the assertion that the socio-material nature of practices in agile draw upon the Manifesto in their formation and continue to rely upon that ideal for their reproduction and maintenance.

Barrett et al., (2013) argue that ideologies, when they are most effective, are deep structures that underpin observable actions but are actually hidden or not accessible to actors. This is quite different to what we have identified at PL, where the ideology is present to all and it is this presence (through the collective cognitive bridge) which we have identified as particularly relevant for sustaining the local socio-material practices. While we do not disagree that often ideologies that underpin actions can be hidden to actors, our findings suggest that in some circumstances these ideologies can and do remain very present. We surmise that this will occur in situations where an ideology translates into local practices that are quite different to what may be considered ‘normal’ or traditional in a particular context. The fact that agile can be a difficult system to sustain, with many organizations only partially and temporarily applying it in practice is testament to the fragility of this practice for software development (Cram and Newell, 2016). In these circumstances, keeping the ideology live through the cognitive bridge can help the ‘unusual’ practices to endure, when otherwise there might be a tendency to revert back to traditional ways of doing things. This was illustrated, for example, with the client who wanted PL to build the software in the ‘normal’ (pre-planned and then executed, waterfall) way. The pressure not to do this was eased by making visible the Agile Manifesto and then talking through the values and principles and how they related to the socio-material practices that they were following.

Our findings also show how these local practices are discussed beyond PL (as in the blogs of PL evangelists as well as the online accounts of people who have been ‘turned off’ by the PL ways of working). While we did not specifically study this here, this public discussion of local practices suggests a recursive link back to the public idea (Nielsen et al 2014), in our case offering more concrete translations of the ideology into practices that can support the ideology, like paired programming. In Figure 9 we portray this as a dotted line since we have not explicitly studied here the impact of this online discussion of PL local practices on the public ideology.

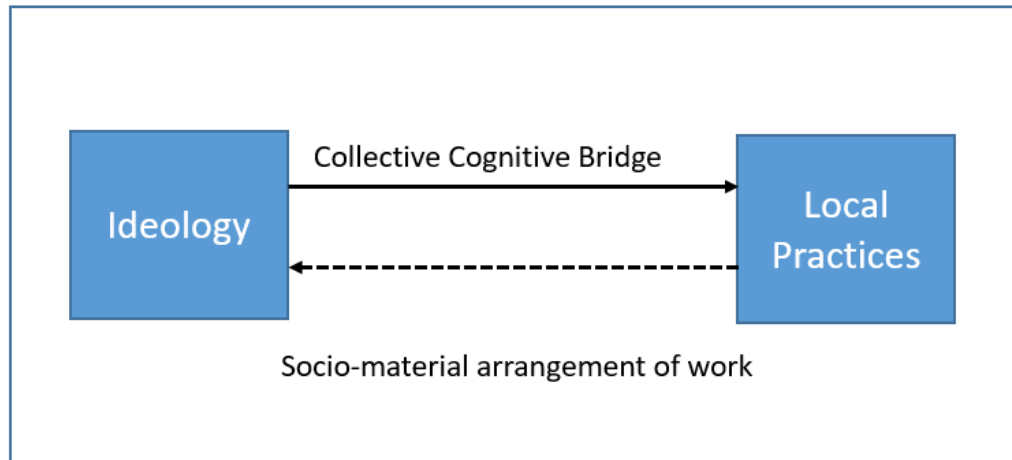


Figure 9: The Role of Ideology in Sustaining Local Practices

The flexibility commonly attributed to knowledge work, and often witnessed in the creative and improvisational performances of knowledge workers, is often generated from a structure that is instantiated in organizational work practices and their integral socio-material systems. Our study illustrates how these practices, which are necessarily local and specific, can draw in an active and persistent way on larger institutional patterns of thought. The socio-material work environment does not so much constrain practice but rather enables actors to get things done in ways that make

sense to them as serving a higher purpose (the agile movement). Power is thus a productive rather than an inhibitive force. From this perspective, the Kool-Aid is good because those drinking it are convinced they are serving a higher purpose.

6. CONCLUSION

Today there is acknowledgement of a recursive relationship between local practices and public ideas, with local practices leading to the adaption of public ideas over time (Nielson et al 2014). But there is little study of how the public idea sustains local practices over the longer term – seeing translation as an ongoing process rather than a one-off process. This paper’s focus on maintenance of local socio-material practices and how these are supported by the public ideology over time, contributes to translation studies as well as socio-materiality research. We identified discourse, material technologies and practices that support work. We linked the work back to the central values of the Agile Manifesto, the seminal document in the agile development community. We introduced the idea of a collective cognitive bridge – the collective role of doctrine, identity, and norms in the ongoing translation of institutional ideas into organizational practices. We suggested that such a cognitive bridge may be particularly relevant in situations where an ideology endorses beliefs and values that are different to other beliefs and values that are present in a particular institutional context.

We explored how the values of the Agile Manifesto relate to arrangement of spaces, technology choices, roles, and activities at PL. We saw, for example, how agile principles are continuously referenced, in relation to identity and norms, themselves playing a part in the maintenance of an active connection between local practice and the public discourse on agile development. Understanding how a public idea can be reiterated and kept sacrosanct and alive,

sheds light on the relationship between broad public ideas and daily operations. In considering possible future directions for research, there is an obvious opportunity, within the current case, to extend our analysis into other domains of activity, such as project management using Tracker.

Our research is subject to the customary limits that go with the study of a single case. It will be helpful to study other organizations that have adopted agile methods to identify whether, for example, the strength of the collective cognitive bridge, influences how far software development is done in a purely agile way. Perhaps, then, where agile is mixed with traditional methods we would see that the collective cognitive bridge is less evident (see Cram and Newell, 2016). It would also be interesting to compare how PL's reliance upon ideology differs from organizations that more readily bring to mind cult-like identities – Twelve Step groups such as Alcoholics Anonymous, religious organizations, or Multi-Level Marketing groups like Beauty Counter. Such a comparison might inform our finding of a more ongoing relationship between ideology and local practice, suggesting when, why and how this relationship persists and similarly when, why and how this relationship, at least in terms of its cognitive accessibility, is severed.

We note also that the elements of any agile environment are not there by accident. Instead, social conventions and material affordances are subject to design, careful selection, and evolutionary modification. This fact points toward research into the genesis of these arrangements and the role of management in bringing them about. The researchers were not present during the founding of Pivotal Labs and therefore have no direct evidence of how the founders of PL applied agile principles to the initial design of the structures, technological affordances, spatial layouts, and business processes of the firm. That agile principles indeed informed those activities is clear from the representations offered on the PL website and the pervasiveness of agile principles in the socio-materials practices at the firm today. Additionally, the authors' empirical observations come

from a relatively small window in time, some years after the firm got its start, and reflect the role of agile ideology in the ongoing reproduction of practices, the discourse about the company, the socialization of new members, and the reinforcement of individual and collective identities.

7. REFERENCES

- Barrett, Michael, Heracleous, L. and Walsham, G. (2013) "A Rhetorical Approach to IT Diffusion: Reconceptualizing the Ideology-Framing Relationship in Computerization Movements." *MIS Quarterly* 37.1 (2013).
- Beck, K., Beedle, M, van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R.C., Mallor, S., Shwaber, K. and Sutherland, J. (2001). *The Agile Manifesto*. Technical report, The Agile Alliance.

- Beck, K. (2010). *Extreme programming explained: Embrace change*. Boston, MA: Addison-Wesley.
- Corbin, J. and Strauss, A. (1990). 'Grounded theory research: Procedures, canons and evaluative criteria'. *Qualitative Sociology*, 1, 3–21.
- Cram, A. and Newell, S. (2016). 'Mindful revolution or mindless trend? Examining agile development as a management fashion'. *European Journal of Information Systems*, 25, 2, 154-169.
- D'Adderio, L. (2011). Artifacts at the center of practices: Performing the material turn in practices theory. *Journal of Institutional Economics*, 7(2), 197–230.
- Gherardi, S. (2012). *How to conduct a practice-based study: Problems and methods*. Cheltenham, UK: Edward Elgar.
- Heracleous, L., & Barrett, M. (2001). Organizational change as discourse: Communicative action and deep structures in the context of information technology implementation. *Academy of Management Journal*, 44(4), 755–778.
- Hultin, L., & Mahrng, M. (2014). Visualizing institutional logics in socio-material practices. *Information and Organization*, 24, 129–155.
- Leonardi, P.M. (2011). When flexible practices meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies. *MIS Quarterly*, 35(1), 147–167.
- Leonardi, P. M. (2012). "Materiality, Sociomateriality, and SocioTechnical Systems: What Do these Terms Mean? How Are They Different?," in *Materiality and Organizing: Social Interaction in a Technological World*, P. M. Leonardi, B. A. Nardi, and J. Kallinikos (eds.), Oxford, UK: Oxford University Press, pp. 25-48.

- Leonardi, P.M. (2013). Theoretical foundations for the study of socio-materiality. *Information and Organization*, 23, 59–76.
- Leonardi, P.M., & Barley, S.R. (2008). Materiality and change: Challenges to building better theory about technology and organizing. *Information and Organization*, 18, 159–176.
- Leonardi, P.M., & Barley, S.R. (2010). What’s under construction here? Social action, materiality, and power in constructivist studies of technology and organizing. *Academy of Management Annals*, 4(1), 1–51.
- Miles, M. B. and Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. 2nd Edition. Thousand Oaks, CA: SAGE Publications, Inc.
- Mutch, A. (2013). Sociomateriality – Taking the wrong turn. *Information and Organization*, 23(1), 25-40.
- Nelson, R. R., & Winter, S. G. (1982). *An Evolutionary Theory of Economic Change*. Cambridge, MA: Harvard University Press.
- Nielsen, J.A., Mathiassen, L., & Newell, S. (2014). Theorization and translation in information technology institutionalization: Evidence from Danish home care. *MIS Quarterly*, 38(1), 165–186.
- Orlikowski, W.J. (2007). Socio-material practices: Exploring technology at work. *Organization Studies*, 28(9), 1435–1448.
- Orlikowski, W.J. (2009). The socio-materiality of organisational life: considering technology in management research. *Cambridge Journal of Economics*, 34, 125–141.

- Orlikowski, W.J., & Scott, S.V. (2008). Socio-materiality: Challenging the separation of technology, work and organization. *The Academy of Management Annals*, 2(1), 433–474.
- Pollock, N., Williams, R., & D’Adderio, L. (2007). Global software and its provenance: generification work in the production of organisational software packages. *Social Studies of Science*, 37(2), 254–280.
- Pollock, N., & Williams, R. (2009). The sociology of a market analysis tool: How industry analysts sort vendors and organize markets. *Information and Organization*, 19(2), 129-151.
- Pollock, N., & Williams, R. (2011). Who decides the shape of product markets? The knowledge institutions that name and categorise new technologies." *Information and Organization*, 21(4), 194-217.
- Rogers, E.M. (2003). *Diffusion of Innovations* (5th ed.). New York, NY: Free Press.
- Ruhleder, Karen, and Brigitte Jordan (1997): Capturing Complex, Distributed Activities: VideoBased Interaction Analysis as a Component of Workplace Ethnography. In Lee, Allen S., Jonathan Liebenau, and Janice I. De Gross (Eds.), *Information Systems and Qualitative Research*, Chapman and Hall. (pp. 246–275). London, UK.
- Stein, M., Newell, S., Wagner, E. and Galliers, R. (2014). Felt quality of sociomaterial relations: Introducing emotions into sociomaterial theorizing. *Information and Organization*, 24, 156-175.
- Swanson, E.B. (1994). Information systems innovation among organizations. *Management Science*, 40(9), 1069–1092.

- Swanson, E.B. & Ramiller, N. (1997). The Organizing Vision in Information Innovation. *Organization Science*, 8(5), 458–474.
- Schultze, U. (2000). A Confessional Account of an Ethnography about Knowledge Work. *MIS Quarterly*, 24(1), 3–41. doi: 10.2307/3250978
- Trans-Situated Learning: Supporting a Network of Practice with an Information Infrastructure
- Vaast, E., & Walsham, G. (2009). Trans-situated learning: supporting a network of practice with an information infrastructure. *Information Systems Research*, 20(4), 547-564.
- Wagner, E., & Newell, S. (2006) Repairing ERP: Producing Social Order to Create a Working Information System. *Journal of Applied Behavioral Science*, 42(1): 40-57.
- Wagner, E., Scott, S., & Galliers, R. D. (2006) The Creation of ‘Best Practice’ Software: Myth, Reality and Ethics. *Information and Organization*, 16(3): 251-275.
- Whalen, J., Whalen, M., & Henderson, K. (2002). Improvisational choreography in teleservice work. *The British Journal of Sociology*, 53(2), 239–258