



**Innovation and the Challenge
of Novelty:
The Novelty-Confirmation-
Transformation Cycle in
Software and Science**

**Paul R. Carlile
Karim R. Lakhani**

Working Paper

11-096

Copyright © 2011 by Paul R. Carlile and Karim R. Lakhani

Working papers are in draft form. This working paper is distributed for purposes of comment and discussion only. It may not be reproduced without permission of the copyright holder. Copies of working papers are available from the author.

**Innovation and the Challenge of Novelty:
The Novelty-Confirmation-Transformation Cycle in Software and Science**

Paul R. Carlile
Boston University School of Management
595 Commonwealth Avenue
Boston, MA
carlile@bu.edu

Karim R. Lakhani
Harvard Business School
Soldiers Field
Boston, MA
klakhani@hbs.edu

WORKING PAPER

Version 3.0

March 15, 2011

**Innovation and the Challenge of Novelty:
The Novelty-Confirmation-Transformation Cycle in Software and Science**

Abstract

Innovation requires sources of novelty, but the challenge is that not all sources lead to innovation, so its value needs to be determined. However, since ways of determining value stem from existing knowledge this often creates barriers to innovation. To understand how people address the challenge of novelty we develop a conceptual and an empirical framework to explain how this challenge is addressed in a software and scientific context. What is shown is that the process of innovation is a cycle where actors develop novel course of action and based on the consequences identified confirm what knowledge to transform to develop the next course of action. The performance of the process of innovation is constrained by the capacities of the artifacts and the ability of the actors to create and use artifacts to drive this cycle. By focusing on the challenge of novelty, a problem that cuts across all contexts of innovation, our goal is to develop a more generalized account of what drives the process of innovation (word count: 168).

Key words: innovation, novelty, artifacts, infrastructure, software and science.

Innovation and the Challenge of Novelty: The Novelty-Confirmation-Transformation Cycle in Software and Science

1.0 Introduction

It was Schumpeter who first breathed modern life into the concept of innovation by distinguishing it from a new idea or invention to a product, process or approach successfully applied to practice (Schumpeter 1934). However, it was also Schumpeter who warned us that the process of innovation itself is indeterminate; for if you could determine the significance of something new at the start then innovation would no longer be a unique and highly valued accomplishment. He went on to state that it was the requirement of novelty that renders the process of innovation indeterminate, and so he considered explaining how novelty is moved from an indeterminate to a determinate state to be the greatest unmet scientific challenge of his day (1932/2005).

In the organization theory and strategy literatures the word novelty is generally used to refer to novel approaches or novel technologies (Zollo and Winter 2002; Winter 2006); so something new that is an innovation (Van de Ven 1986) or something new that is a competitive threat or disruptive (Porter 1998; Christensen 1997). So whether a novel approach is good or bad depends if you are the one creating it or having to compete against it. What has not been discussed in the literature is that while innovation does require novelty, the challenge is that not all sources of novelty lead to innovation, so the value of those sources must be determined. In biology, evolution requires mutations, but most mutations are not viable when they emerge and so are selected out (Ridley 1993). In social evolution the state of indeterminacy that Schumpeter referred to arises because determining value stems from existing knowledge that people use which often creates competency traps (Leavitt and March 1988) and barriers to innovation (Carlile 2002). These barriers are only addressed when particular knowledge that people use to determine value is changed in a specific way. Without such changes in how value is determined it is not surprising underperformance when it comes to trying to innovate (Sorenson and Stuart 2000).

We propose that the challenge of novelty is addressed by what we call the novelty-confirmation-transformation (N-C-T) cycle. For an individual then innovation is a cycle that requires both a capability to develop novel courses of action and a capability to confirm their value. This effort establishes a sweet spot for innovation where the consequences identified help

an individual determine what knowledge to transform and what to keep the same to develop the next course of action to drive the innovation cycle. However, most innovation involves more than one person; often many individuals specializing in different domains, so the question that arises is how do actors distributed across different domains address the challenge of novelty collectively?

To address these socially distributed requirements we empirically examine what we call an “infrastructure for innovation.” Infrastructure is made up of two things: artifacts and actors (Star and Rehleder 1996). By artifacts (Star 1989; Carlile 2002) we refer to “human constructions” that range from material artifacts such as prototypes and measurement tools to more symbolic ones such as “language” and shared methods that reference and coordinate action with material artifacts or other actors (i.e., speech acts; Searle, 1969). What all artifacts share in common is that they are used by actors to carry out their actions with other actors. Artifacts naturally vary in their capacity to represent novelty and confirm its value depending on the circumstances in which they are used (Carlile 2004). By actors we refer to individuals and their abilities to use artifacts to develop and perform a particular course of action. In the case of innovation those abilities are to develop novel courses of action, confirm or determine their consequences, and based on those consequences transform existing knowledge to develop additional courses of action that can further drive the cycle of innovation. Overall, an infrastructure’s capability for innovation is determined by the capacity of the artifacts and the abilities of actors to develop and use artifacts to drive the innovation cycle. If an infrastructure is lacking particular capacities and abilities it will be limited in its potential for innovation by comparison to an infrastructure that has those capacities and abilities. A description of the relative performance of an infrastructure for innovation provides one means of outlining the sources of organizational heterogeneity (Chandler 1962; Henderson and Cockburn 1994).

To fully conceptualize how an infrastructure for innovation can address the challenge of novelty we have chosen to empirically present two very different contexts of distributed innovation: one in the computer sciences and the other in the life sciences. In the computer sciences we examine an open source software (OSS) development community where the artifacts used (i.e., computer code, compiler, test suites) are quite stable and “globally” shared among the actors involved. This makes addressing the challenge of novelty reasonably straightforward. In the life sciences we study an open collaboration of scientists developing treatments for Multiple Sclerosis (MS) called the Myelin Repair Foundation (MRF). Because so little is known about

what causes MS there is significantly more novelty associated in developing innovative outcomes than in software. Further, the actors who are working together in this new scientific collaboration are highly specialized and there is little common language or metrics to share and assess the value of the knowledge being developed. In the first setting existing capacities and abilities are used to address the novelty. In the second setting new infrastructure (capacities and abilities) have to be developed to address the challenge of novelty associated with MS.

From a theory building perspective an examination of extremes is useful because one can identify the attributes of how an infrastructure facilitates the N-C-T cycle in one setting and compare it to the attributes of how an infrastructure addresses it in a more complex set of circumstances (Christensen and Carlile, 2009). Such a comparison allows us to begin to move from “descriptive” theories of innovation to more “formal” theories (Glaser and Strauss, 1967: 35).

In what follows we will outline the challenge of novelty and then introduce the novelty-confirmation-transformation (N-T-C) cycle to conceptually explain how the novelty challenge is addressed. We then discuss how the N-C-T cycle occurs within infrastructure for innovation consisting actors with particular abilities to develop and use particular artifacts to increase and confirm sources novelty. We then present our two cases that demonstrate the operation of our framework for two very different set of innovation circumstances. We conclude the paper by proposing that the framework presented here provides a conceptual and empirical basis to develop a general theory of innovation; understanding what is required to address the challenge of novelty across a variety of innovation circumstances.

2.0 Opening up the Black Box of Innovation

Schumpeter struggled to adequately describe the process of how the challenge of novelty would be addressed, eventually describing innovation as a process of creative destruction (1935). In a similar vein those who followed him labeled this process “creative abrasion” (Leonard-Barton 1995) or a “dynamic capability” (Teece, et al., 1997). Such descriptions imply creation and change, but the process of determining what to create or what to change remains black boxed. Innovation has also been described as a process of combining old and new knowledge (Schumpeter 1934; Winter 1983; Galunic and Rodan, 1998). New knowledge gets at the requirement of novelty and old knowledge gets at a potential means of determining value, but a process of recombination doesn’t specify how the natural tension between old and new is resolved; so addressing the challenge of novelty remains. This situation requires a process where actors

develop sources of novelty, confirm the consequences of that action, transform existing knowledge and then develop another course of action to continue the cycle of innovation.

2.1 The Novelty-Confirmation-Transformation Cycle

Framed in terms of novelty, innovation is a challenge because an actor must develop sources of novelty while at the same time use existing knowledge to determine the consequences or the value of that novelty for innovation to occur. So to make progress we move beyond the tension of new and old knowledge and reframe it in terms of developing sources of novelty and confirming its value. Novelty is a reminder that not everything new is of value and confirmation is a reminder that not everything old is bad. So the black box is more than just a process of recombining old and new, more than a tension of old and new, but a process of creating a novel courses of action, confirming its value based on the consequences identified from it use and then transforming one's knowledge accordingly to develop another course of action.

To specify this process more fully let us consider the extremes of novelty and confirmation (see Figure 1). Under conditions of 100% confirmation any source encountered confirms and so conforms 100% to previous approaches or patterns (Jantsch, 1980). Under these conditions innovation is not possible because nothing new or novel can be valued by an actor. To push in the opposite extreme under conditions of 100% novelty any source encountered lacks any confirmatory pattern and so no potential value can be appropriated by an actor. Under these circumstances a state of chaos exists and innovation cannot take place. Figure 1 represents all the potential circumstances faced by scaling them starting at the extreme of 100% confirmation (so no potential to identify sources of novelty) and then moving out to conditions of 100% novelty (so no potential to confirm the value of novelty). So conceptually speaking the novelty, confirmation and transformation cycle occurs at a “sweet spot” of innovation that lies somewhere between these extremes.

<insert Figure 1>

However, this conceptual representation must give way to a practical one that expresses empirically the distributed challenge of establishing a sweet spot of innovation. By distributed we mean that innovation is more than just individual creativity, but is a collective effort of actors often across different domains or thought worlds (Dougherty 1992). These differences often beg the question if there is a shared knowledge or common language for actors to use as they jointly assess the value of novelty. Establishing this common language is complicated by the fact that actors

have existing means of valuing novelty within their domain. Figure 2 expresses this relational or social dimension by reflecting a mirror image of the sweet spot of innovation in Figure 1. This indicates the work necessary to develop and confirm the value of novelty and transform knowledge at this relational “sweet spot” for innovation. Here the capacity of the artifact that supports this “sweet spot” could be described as a boundary object (Star, 1989; Carlile, 2002). But as is often the case artifacts that worked as boundary objects in the past may have to be changed to establish new capacities to develop and value new sources of novelty. Even if effective boundary objects are present the innovation cycle can breakdown if one actor lacks the ability to use the capacity of the artifact to represent novelty or identify consequences associated with it. Thus a breakdown in the capacity of the artifact as boundary objects or the ability of the actors using it further reveals the relational properties between an artifact and actors (see Figure 2) that must be met when identifying novelty and determining its consequences when the task of innovation is distributed across domains (Black and Carlile 2009; Carlile, 2004).

<insert Figure 2>

Overall, the distinction between confirmation and novelty offers greater causal specificity of the tension between what is old and new because confirmation refers to a particular “valued” outcome generated by a previous pattern; not just something old. Confirmation then specifies the value (and how it is structured) of what anchors the old, and so it concretely problematizes its relation to what is novel (i.e., capacities to represent, abilities to use current knowledge and associated interests). Further this process is an ongoing cycle where some novel courses of action are confirmed while others are disconfirmed which helps actors determine what knowledge to change and what to keep the same to develop the next course of action.

2.2 Infrastructure for Innovation

But how is this done and how can we empirically describe this innovation cycle? So far we have paid particular attention to the empirical categories of artifacts and actors. Both these categories have strong traditions in Science and Technology Studies (STS) literatures; in actor network theory (Latour and Callon 1995) and in Star and Ruhleder (1996) concept of infrastructure. We adopt the concept of infrastructure because it emphasizes the relational performance of structure as both social (actors) and technical (artifacts) and is inherently distributed (Gal, Yoo and Boland 2004). Infrastructure also connotes how structure can be become taken for granted until breakdowns in performance reveal specific parts of it (Star and Ruhleder,

1996: 113). Where we extend Star and colleague's notion of infrastructure is that while they are primarily focused on describing an infrastructure, we proceed further to compare how a given infrastructure differs from another in terms of its performance in our case of the N-C-T cycle.

More specifically by comparing performances we can generate insight about what determines the relative performance of an infrastructure in terms of generating novelty, confirming its value and transforming knowledge based on the consequences identified. This is what leads us to contend that the performance and so the capability of an infrastructure for innovation is determined by the capacity of the artifacts and the ability of the actors to develop and use artifacts to drive the novelty-confirmation-transformation cycle.

But how do actors use artifacts to concretely go through this cycle of developing innovative courses of action? Consider the individuals who have developed new "extreme" sports such as mountain biking, wind surfing and snowboarding that have been discussed a great deal in the innovation literature (Franke and Shah 2003; Hippel, 2005; Schreier and Oberhauser, 2007; Baldwin and Hippel 2009). In the beginning stages these actors take existing artifacts such as a bike, surf board or skate board and modify its use in an activity in a new way. In the case of windsurfing a key effort was where and how to attach the sail to not only get good power, but also balance and maneuverability on the deck. The use of these artifacts led to more modifications about the location and size of the sail and how to "rig" it to the board in such a way that still allows the user to move around the sail easily as they shift their weight. This is clearly a process of trial and error, but if we look more closely actors put some artifacts into use in an activity in a new way and based on the consequences observed they intentionally transform some artifacts to improve performance in the next cycle. Sometimes those changes improve performance and sometimes they do not; so confirming or disconfirming the performance of a course of action is essential to the N-C-T cycle. And so with the development of artifacts in a new way (a novel course of action), confirming what worked better and what didn't, actors modify their artifacts and then put them to use again and the innovation cycle is repeated. At the most basic level novelty or newness is created by actors through some artifacts. These artifacts may be physical or they may be symbolic (i.e., language as speech acts; Searle, 1969) and so indirectly refer or effect other artifacts or how actor use or change a particular artifact.

An historical example of how an infrastructure of actors and artifacts is developed to perform better relative to an existing one can be seen in the development of the Kanban approach

that became the revolutionary building block of the Toyota Production System. The Kanban system was a reaction to the waste and lack of adaptation created by the traditional push manufacturing system that characterized the majority of production systems in the first half of the 20th Century (Fujimota, 1988). For push systems efficiency was defined by maximizing individual machining and manufacturing processes by pushing throughput. The result of this was high inventories between individual operations and a large number of defects or latent problems hidden in those inventories. Defect rates remained high in push systems because each defect as a potential source of novelty was hidden in large inventories and once identified the costs and time frame associated with confirming the root cause in order to address the problem were very large.

In the resource scarce world of post World War II Japan this level waste was unworkable for Toyota and so they implemented a pull manufacturing system build around a Kanban approach. Here the production of parts in the first step is triggered by the consumption of parts in the second. This pull system was established through a combination of part bins and Kanban cards that specified the number of parts that should be in the bin; creating a very clear dependency between actors. The use of these “pull” oriented artifacts made defects as a potential source of novelty, as indicated by more or less parts than specified by the Kanban card, easy for actors to identify and then confirm courses of action to address root causes. It is this pull infrastructure that became the backbone of continuous improvement at Toyota.

A push production system also has an infrastructure, but it is much slower to confirm defects and develop courses of action to drive continuous improvement. Change and transformation does occur in a push system, but its infrastructure lacks the capability to make such changes timely or well targeted at the root cause. The overall historical difference between a push and a pull system is that the latter’s infrastructure has a greater capability to identify sources of novelty and more easily confirm what courses of action will drive continuous improvement (Fujimota, 1988; Young 1992; Spear and Bowen, 1999). The key take away from this comparison of a push and pull system is that infrastructure is all around us and is for the most part taken for granted; and so as we consider how to do innovation on a broader scale we must reveal it’s ubiquitous and taken for granted nature to better enable the potential role that actors and artifacts play in addressing the challenge of novelty.

3.0 Overview of Cases

To demonstrate how the novelty-confirmation-transformation cycle operates in an infrastructure for innovation, we use two cases that are in the domains of software development (software) and drug development (science). The first case highlights how a massively distributed community of software developers seamlessly uses an *existing* infrastructure to develop a robust open source software (OSS) database project. OSS development is a relevant case study because the relatively fluid organizational boundaries of OSS projects make them susceptible to increasing novelty due to the constant influx of newcomers on projects (von Krogh, Spaeth and Lakhani 2003). These newcomers constantly bombard (successful) OSS projects with bug reports, new feature requests and software patches and updates based on their own use and create significantly more novelty than similar projects that might be proprietary.

The second case highlights the development of a *new* infrastructure for innovation in pursuit of an elusive goal, a treatment for Multiple Sclerosis. Drug development is fraught with uncertainty and ambiguity with the average result of most traditional means of organizing research and development in this domain resulting in failure (DiMasi et al., 2003). This case shows the response of one non-profit foundation, the Myelin Repair Foundation, to re-conceptualize the organization of scientific work in basic drug discovery to accelerate the creation of a treatment for MS. Here the challenge of novelty was multi-fold; including; new actors collectively working across disciplinary boundaries and new work arrangements that emphasized interdependencies amongst actors, their scientific laboratories and institutions and disciplines.

The logic for presenting the two very different circumstances (the first straightforward and the second more complex; the first utilizing existing infrastructure and the second developing new infrastructure) is that these differences allow us to tell in the case of OSS a very detailed story of how an existing infrastructure is used to address the increasing novelty and in the more complex case of MRF focus on key moves that actors took to develop a new infrastructure to address the high novelty and low confirmation circumstances faced. This “stretching” and the key differences in circumstances that follow allows us to demonstrate the potential value of the framework being presented as we push toward outlining a more general theory of innovation.

The cases are based on primary data collection; with the OSS case developed through real-time tracking, observation, and key participant interviews of an unfolding OSS project encompassing a complete software development cycle over a one-year period of time. The MRF

case developed over a four year period that consisted of interviews and observations of all the key actors with access to proprietary documents and attendance at several key strategic scientific meetings amongst the principles. The level of detail presented in each case also provides insight into the infrastructure of innovation available in each effort; the OSS infrastructure can be observed and analyzed with high precision due to the extremely high fidelity of the artifacts used in the creation of the software. Access to email lists, source code repositories and software defect lists enable us to track to source code at the individual line level, the origins of novelty, the confirmation approach taken and the resulting transformation of the source code. This allowed us to highlight the basic events and mechanisms underlying the novelty-confirmation-transformation framework. Due to the complexity of drug development and the relatively long time frames taken to create a drug (typically 15 years from start to finish (DiMasi, et al., 2003)), the MRF case provides an overview of the key moves in developing this new infrastructure and explore its operation and effectiveness via an illustrative vignette to demonstrate the infrastructure in action. Figure 3 provides a comparison to scale how the “sweet spot” of innovation differs across the two cases and the Toyota example presented earlier. The figure illustrates how challenge of novelty increases across these three settings.

<Insert Figure 3>

4.0 A Ready Made Infrastructure for Innovation: Open Source Software

The relative success of open source software (OSS) development projects has been an enigma for both scholars and practitioners of innovation. Over the past decade, academics in a variety of technical and social science disciplines have tried to build our understanding of the emergence, operation and success (or lack thereof) of OSS communities (Roberts et al, 2006; von Krogh and Hippel 2003). That complex software systems running mission critical applications can be designed, developed, maintained and improved in a distributed setting, outside of the boundaries of traditional firms, with participants from around the world with heterogeneous motivations and organizational affiliations has been a major driver of scholarly inquiry. Equally interesting is that some of the largest software companies and the biggest holders of intellectual property (e.g., IBM, Google, Apple, and Oracle) have embraced OSS communities by encouraging the participation of their own personnel in, and donating copyrighted software and patents to, these communities, and integrating OSS software into their strategic product and service offerings.

While a vast majority of OSS projects are relatively small (Krishnamurthy 2002), with a handful of community members devoted to solving specific technical problems, there are a significant and growing minority of projects (for example: Linux, Debian, GNOME, Apache, Perl, Mozilla, Hadoop) that have achieved large-scale mobilization that involves hundreds if not thousands of programmers contributing code and working collaboratively. Much of the existing scholarly attention on these large-scale projects has primarily been around answering macro questions related to the puzzle of private investment for a public good (von Krogh and von Hippel, 2003, Murray and O'Mahony), motivations of participants (Lakhani and Wolf 2003; Roberts et al, 2006), membership structures (von Krogh et al 2003, O'Mahony 2003), governance regimes (O'Mahony 2003; Weber, 2004) and the role of technical architectures in enabling distributed work (MacCormack et al., 2006). While the existing literature helps us understand the institutional arrangements and technical choices that provide the pre-conditions for large-scale mobilization, it does not address how increasing sources of novelty via new member participation is addressed in these open communities. We build and extend this body of knowledge by focusing on the microstructures of interactions amongst distributed actors as they work collectively via a variety of artifacts to solve innovation problems.

By all accounts, software development within the structures of any one organization is difficult enough with persistent and significant issues related to quality and timely delivery and failure (Shen et al 2000). And one should expect that these problems are potentially greatly exacerbated in a distributed open community. In particular, OSS communities have to account for the input of multitudes of distributed contributors, working asynchronously on their own technical issues and absent “formal” managerial hierarchies driving alignment, and transform them into a coherent technical product that exhibits high performance and quality.

We posit that the core to the success of large-scale OSS communities is the use of a “readymade” infrastructure for innovation that enables the addressing of increasing novelty. The artifacts constituting the infrastructure are relatively well-established and mature technologies like broadcast email lists, common compilers and standardized test suites, and code repositories. These artifacts were not specifically designed for the task of distributed open source software development, rather they are general purpose communication and computation tools that were repurposed and adapted in service of global communities working together.

E-mail lists are the primary means of sharing information and coordinating work in OSS communities. A vast majority of technical activity occurs via contributors posting and replying to messages on the various public email lists of the community. This one-to-many communication flow ensures that all interested parties have at a minimum the ability to access the key issues at hand in the community. All conversations between individuals are public conversations as discussion on the e-mail list are sent to all subscribers and archived on public websites. The public nature of the conversation and the ability of “anyone” to chime-in to an ongoing discussion mean that the development of novelty in the community through new members and the identification of new issues are immediately made visible and available to all other community members. Thus the burden of developing novelty and understanding its consequences is easily shared amongst the community members.¹

The e-mail list is the “place” where discussion of the software development activity for the community occurs. E-mail messages become the medium (work happens inside of an e-mail message) and/or carriers (actual code or signals of code completion) of all the community work. The list then broadcasts all the work to subscribers and provides a continuous signal to the entire distributed membership as to current state of affairs in the community. The act of working and participating in the community becomes the act of communicating as well, and so the artifacts for doing the work and communicating about the work are not separate. The email list captures the essential task of an innovation infrastructure; novelty arises through the email list through new issues and ideas being posted or new people joining the list. Confirmation occurs through the interactions amongst the users on the list as they try to assess if the novelty presented is of consequence. Finally the email list is the vehicle for transformation as changes that are under consideration are communicated, debated, discussed and finally enacted through that medium.

All community members use a common software compiler to run and assess the code they are using and potentially change it. Compilers are used to create and change high level computer languages (e.g.: C, C++, Java) to low level machine instructions. The compiler is the primary means by which transformation occurs in the infrastructure. Individuals, based on their own needs or responding to the discussion on the email list will create or modify existing code to meet the emerging consequences as they begin to move between the requirements of novelty and

¹ This of course presupposes that the communities actually mobilize many developers – which may not be the average outcome for most open source projects.

confirmation. The compiler is used by individuals to test and verify that the code developed actually works as intended; so serves a very strong confirmation purpose. The compiler is also essential to community members because they can confirm that contributions from others work as advertised and also identify potential areas of weaknesses or bugs based on the compiler reports. For all of these reasons the compiler serves as the community's "village idiot", a globally shared standard, which is hard to argue against given the standardized ("objective") reports produced by the compiler.

The compiler and related test suites enable confirming the value of completed work, i.e. "Does it do what the contributor claims?" via "objective" measures. This verification works at both the basic level, i.e. does the changed system actually work as advertised, and at a systems level, i.e. is there any degradation in performance (speed, load handling, etc.) due to the changes. Software development kits provide standard tools that produce shared ("objective") measures about the consequences of software changes, which can then be easily shared with others and serve as a basis for further development and innovation. These shared standards allow those submitting problems to utilize an "objective measure" to make their claims and provide detailed information on how specific problems were created along with associated computer-generated log files and use scenarios which allow others to recreate those sources of novelty on their own local machines.

The work produced by actors' in an open source community—the software code—resides in a public repository referred to as a software version control system. The repository keeps track of all changes made to all files and enables roll-back to previous versions if an error is found in the most current update. The most commonly utilized system in open source is called a concurrent versions system (CVS). A key feature of CVS is that it does not lock files that are currently under development. Thus any authorized user of CVS can change any file – regardless of the status of that file in the community. This increases the potential for novelty to be developed. A vast majority of open source communities configure CVS so that anyone can anonymously download the software code they are interested in – with only authorized members allowed to make their changes "official." In addition all changes made to the CVS are broadcast on an email list to any interested member. The integration of the repository with the email list provides an additional means of both broadcasting the novelty that is being introduced into the system to the entire community and the potential to identify its consequences. CVS as an artifact has the capacity to coordinate the collective development of the code; it is used to store the code; it maintains a

history of all the changes; it can be rolled back to an older stable versions if problem arise and it integrates with the email list to continuously inform members about changes made to the code.

Unlike a firm-based system for software development actors in open sources software communities self-select into programming tasks of interest based on their needs, expertise and interest. Typically newcomers to an open source community join not on the basis of credentials, but by accomplishing specific tasks in the software development cycle (von Krogh et al 2003). This self-selection means that individuals join and participate when they feel confident that their input and contributions will be of sufficiently high quality so as to warrant attention and effort by other community members. The open email list subscription policy means that members are the primary source of novelty by bringing forth problems they face in their work or developing new functionalities in the code. Established community members engage with newcomers in a joint process of confirming the value of the novelty they developed and then transforming the knowledge and code they have used to address consequences identified. Established members in the community also develop novelty or assist others in a process of confirming their value and where consequences are identified transforming code to address them.

4.1 An Open Source Problem Solving Vignette: Developing a New Complex Functionality in a Distributed Fashion

We illustrate the use of an infrastructure for innovation in open source by providing a vignette of the development of new software functionally in the PostgreSQL database community. The PostgreSQL (PG) software and community is one of the early open source software initiatives with a lengthy evolution and a history going back to the middle of 1980s. PG development occurs outside the boundaries of any “formal” organization². The community is virtual in both the literal and physical sense and there is no formal organization driving the development of the software and there is no physical space where the developers work together. PG by itself does not have any employees. However, some developers do have jobs that allow them to work on PG software – full time and part-time. On PG’s website there are references to the “PG Global development group” however this is more for marketing purposes and does not really serve any legal or administrative function. A volunteer steering committee of five members, most of whom were part of the original effort to organize users, handles public relations, decides on official software releases and provides

² Some open source projects have incorporated as non-profit organizations (for legal protection purposes) and often have a formal board of directors.

“write” access to other members for the CVS repository, i.e. they can make official changes to the PG software code. The community congregates by using 22 e-mail lists related to the various aspects of the project. Majority of the technology development activity occurs on the “hackers” and “patches” e-mail list. All of the lists are open except for one which is reserved for five members of the PG steering committee.

The development and integration of the Internet Protocol version 6 (IPv6) feature provides a demonstration of the way in which an infrastructure for innovation supports massively distributed collaboration in OSS and how increasing sources of novelty are handled. IPv6 is a relatively new internet protocol that provides enhanced security and more efficient data transmission capability over existing and the more commonly used IPv4 protocol.³ These network protocols are the core means by which the database communicates with its clients and thus need to be significantly robust.

We tracked the development of this feature by first looking at the release notes of the latest version, at the time of data collection in 2004, of PG (version 7.4) that listed new features developed since the last release. We then worked with one community member to identify all the software code changes made in the CVS repository that enabled the development of particular functionality in PG. The dates and times of the software changes then provided clear markers to track the email discussion lists and to uncover the process by which the software code got developed. We followed up the email reconstruction with phone and email interviews with individuals who played a key role in its development. Overall the IPv6 feature was fully developed over a 10 month period and involved 17 discrete changes to the source code (adding 903 and removing 584 lines of code) and involved the participation of 34 individuals from 11 different countries. Actual coding for the feature was done by eight individuals at different times of the development cycle. No other commercial databases supported IPv6 at the time of PG’s release.

Most surprisingly the IPv6 functionality arrived as a novel but “pre-made” solution to the PG community on November 28th 2002. Up till this point there was no stated goal in the community to build in IPv6 functionality within the software code. N. Kukard from South Africa announced to the development list that he along with a colleague had adapted the reference IPv6 implementation and code from Japan, for an un-related project, to the current version of the PG

³ Most computers use IPv4 protocol for communication over the network.

software and that they wanted to donate this code to the community. The Japanese implementation was a collaboration of 13 organizations (university-firm partnership) to develop and implement the IPv6 specification in open source operating systems. Kukard and his colleague adapted the Japanese implementation to work on the PG database to meet their particular use need of supporting secure financial transactions and had not previously participated in PG development. They participated by submitting a code “patch” to the developer email list which had all the changes that needed to be made to the PG software base for the IPv6 functionality to be enabled.

Figure 4 highlights the iterative cycles of novelty, confirmation and transformation undertaken by the distributed community of PG developers once the IPv6 patch was received. The 34 individuals who participated in its creation played a variety of roles during the production process including; code writer, reviewer, discussant, tester, integrator and committer to CVS. Kukard’s initial code submission got immediate attention on the developer list when a non-core periphery member⁴, N. Conway, immediately noted variance from coding style conventions between the PG community’s expectation and the submission. This type of confirmation, although on the surface appears trivial, is important to ensuring that the diverse individuals participating in open source development have a common reference style for understanding code. Equally important to note is that the source of the confirmation was a periphery member who had recently started to participate actively in the developer community. Indicating that the openness of the PG community’s infrastructure enables newcomers to make substantive contributions to novelty and confirmation relatively easily and quickly in areas they have interest and ability.

<insert Figure 4>

Kukard immediately transformed the code and resubmitted it to the developer’s list based on the style and formatting critique. The code then got further transformed by a core developer, B. Momijian, who further modified the code to fit within the structure of the PG architecture. Here we observe an initial novelty, confirmation and transformation cycle for this feature. However, this is one of the many cycles that are instantiated within the PG developer community infrastructure and it highlights how a very complex technology gets developed and put into practice by the diverse individuals that are participating in its creation. A majority of novelty

⁴ Consistent with Von Krogh et al (2003) a core member of an OSS project is one that has CVS write access rights, i.e. the changes they make to the repository are considered official, and periphery members lack such access.

related to IPv6 is introduced through users finding problems with the latest version of the code⁵ as they try to run the database within their own heterogeneous use environments and then either attempting to create a local solution to the problem themselves and/or alerting the developer email list of the problem for general consideration and help.

The novelty generated is due to the newness of the IPv6 technology and its integration with a mature software system that runs on many different systems and conditions. Ex-ante, the developer community does not have a full understanding of the changes required to adapt both the IPv6 patch and their own systems to robustly enable the feature within the PG software. Overall we observe 13 episodes of novelty during the development of this feature. Novelty development and identification is immediately followed by confirmation where more established user/community members identify consequences and pinpoint their source. Here the CVS and compiler artifacts provide significant assistance in tracking the exact cause of the problem and address them by transforming specific parts of the code.

The fact that this occurs in a public email list means that initial diagnoses about the source of the novelty have a higher probability of being confirmed or disconfirmed since it is easy for a large number of people with a larger variety of approaches to participate. Hence, immediately following B. Momijian's initial commit of the IPv6 functionality, J. Conway, another periphery member found the latest CVS version of the software not working at all in his computing environment. Joint diagnostic work with B. Momijian revealed that this problem could be tracked down to the recent changes made in the code for IPv6. Momijian solved the initial problem by simply reversing the CVS commit and restoring the code to the older state. He then used that interaction to guide the further changes he made to the IPv6 patch which he then recommitted to CVS two weeks later. The IPv6 feature development had 20 confirmation episodes.

Transformation in the innovation infrastructure comes about after confirming the value of the novelty. Here the object of innovation, the concrete artifact (e.g., software code) that is being developed, is being changed to reflect the new understanding about the consequences associated with a given sources of novelty that must be accounted for to satisfy the performance the requirements of the software. Transformations itself then leads to more novelty that can then repeat the cycle we have outlined. In our setting there were 17 episodes of transformation for this

⁵ In most open source projects once a feature is made committed to CVS it becomes part of the alpha distribution of the code. This means that all developers and users interested in advanced early functionality will use the latest CVS version of the code to run their systems. This is prior to the code entering beta-testing or becoming an official release.

one feature. The above analysis also reveals that there is not necessarily a strict one-to-one ratio of novelty-confirmation-transformation. While novelty events trigger the cycle, confirmation and transformation can occur multiple times within a cycle until a resolution is achieved.

What is striking about Figure 4 is the number and diversity of individuals participating in the co-creation of the IPv6 functionality. While the initial work on the IPv6 was done in an unrelated project and the first application to PG occurred via an individual who was not a core member, indeed Kukard stopped participating after sending in the patch, other members of the community jointly took over the further development and integration of this feature into PG. But there was no active management of the process. The high confirmation capacity of the artifacts used to coordinate and produce open source software and the self-selection of interested individuals participating in the community constitutes an infrastructure for innovation that can eschew traditional managerial intermediaries and directly pursue cycles of novelty generation, confirmation and transformation. This infrastructure is narrowly focused on solving technical problems surfaced by their users and so there are no concerns about market share, sales strategy, marketing, technical support, competition etc. Indeed open source communities thrive by ignoring these features or relegating them to other complementary communities. This focused and readymade infrastructure provides a capability to develop novelty and confirm its value. And this establishes a sweet spot where the consequences identified help actors determine what code to change and what to keep the same.

5.0 The Myelin Repair Foundation (MRF): Creating a New Infrastructure

As early as the middle of the 1990s, the drug discovery and development process responsible for finding cures and treatments for serious diseases and illnesses, as practiced by most major pharmaceutical firms, was incurring a significant “innovation deficit” (Drews 2003). Despite huge investments of both public and private monies, increasingly sophisticated tools, technologies, and techniques, and the advent of genomic sequencing, proteomics, systems biology, high throughput screening and other marvels, R&D output of the pharmaceutical and biotechnology industries, as measured by the number of novel drug applications to the FDA, was considered “low,” while costs continued to skyrocket—some \$50 billion annually (Munos 2009). Several analyses suggested that the industry was in the midst of a productivity crisis with flat to declining approvals of new molecular and biological entities and a steady year-over-year increase in R&D expenditures (FDA White Paper 2004; Cockburn 2006). The latest research showed that

more than 10 to 12 years and \$1 billion were needed to bring a drug, from the laboratory bench to the patient bedside (Munos 2009).

Many reasons have been cited for this apparent drop in performance, including the idea that the ability to systematically innovate in the past was attributable to pursuing single-cause diseases. Thus, the vast increase in lifespan occurring in the first half of the 20th century, especially in the developed world, was driven by relatively low-technology interventions in combating infant mortality and relatively simple (in hindsight) drugs aimed at combating infectious diseases and vitamin deficiency, along with public health improvement projects like clean water, sanitation, and pest control (Epstein 2006). Many high-impact innovations, like ether for anesthesiology, penicillin for infections, and insulin for diabetes, came about because it was possible to identify a single-cause biological mechanism and control it by relatively simple (in retrospect) chemical entities. However, the current organization of the R&D work dedicated to this “low-hanging” fruit innovation strategy was poorly suited to identifying and addressing the multi-causal nature (i.e., genetic, environmental, infections) of the remaining disease categories (i.e., neurological, cancers, etc.).

Meanwhile, the number of academic papers published in the life sciences had increased by a factor of eight in the past five decades, exceeding 800,000 *annually* by 2008 and appearing in over 5,000 journals⁶. Such vast expansion in knowledge might have suggested there would be greater, and faster, results in treatment and cure development. But paradoxically, the opposite had occurred. Scientists had begun to narrow their focus of investigation and were measured and rewarded for “publishing results” in specialty journals—a process that could take anywhere from 3.5 to 5.5 years from discovery to publication (Stern and Simes 1997) and that generated even more and wider gaps in knowledge across disciplines (Dougherty 2007). The published research, moreover, was typically far removed from the practicalities of drug development, and hence from “bedside” results. Furthermore, while academic science publicly and historically espoused the values of openness and knowledge sharing, the reality was that most scientific labs were highly secretive and rarely collaborated and shared knowledge with others *before* and even *after* the publication of their results (Campbell et al. 2006) thus reducing the possibility of collectively solving tough problems.

⁶ Data compiled from <http://dan.corlan.net/medline-trend.html>

Given that productivity is stagnant and the diseases remaining are more complex why aren't new approaches to drug discovery being systematically tried by those in the industry? Currently expensive mergers targeted at acquiring the pipeline of "developed" new molecular entities are the primary approach being used to deal with this innovation crisis (Business Week 2009). The MRF stands apart as a new approach to drug discovery by doing scientific work across disciplinary and organizational boundaries to develop treatments in the multi-causal disease area of MS. The MRF provides a unique opportunity to look at circumstances where new infrastructures have to be developed to address the significant challenge of novelty inherent in a complex disease.

The impetus to creating the MRF was rooted in its founder's, Scott Johnson a former management consultant and successful president and CEO of three startup companies, own experience as a patient suffering from MS. Johnson was diagnosed with the disease in 1976 as a young adult and was told that a cure would be ready in about 25 years. Unfortunately there was no cure and in 2004 Johnson decided to take matters into his own hands by forming a non-profit organizations geared towards easing the suffering of MS patients through a cross-disciplinary and cross-institutional basic research collaborative. By 2010, MRF had funded more than 45 research projects, most of the type that organizations that typically underwrote research (like the National Institutes of Health) would deem "too high risk." The work generated 18 patentable inventions, the first patent being awarded in 2008, along with 27 new research tools essential for conducting sophisticated research in neurological diseases. MRF scientists published their findings in more than 60 peer-reviewed journal articles. All told 19 new potential myelin repair drug targets had been identified and six were being advanced to the next stage of validation. Johnson, reflecting on the achievements stated: "by most benchmarks, the scientific results of the collaboration were above and beyond the expectations of both skeptics and supporters of the MRF."

So how do you create or more accurately what are the moves to build new infrastructure in such a challenging scientific area where significant gaps exists (Dougherty 2007)? The first move goes right at the need to create an initial sweet spot where the N-C-T cycles associated with this multi-causal disease can be more easily driven. Given the multi-causal nature of MS (i.e., genetic, environmental, autoimmune, viral) there is much that scientists don't know about treating the disease and much still needs to be identified and explored from and across a variety of disciplinary backgrounds. Without the development of shared artifacts (i.e., shared hypothesis, common testing criteria, agreed upon dosage rates, shared data, etc.) confirming the value of these sources

of novelty will be slow in coming and hard to leverage given their isolated development. This is the troubling situation when a significant innovation gap or challenge exists; you need sources of novelty but you also need to develop new approaches to confirm the value of those novel sources. To address this sizable paradox the MRF choose to go against a primary scientific starting point that has dominated both academic and for-profit science firms especially when we consider the funding of that science. The MRF took a key insight that came out of the development of an AIDS treatment and decided to move away from the goal of curing the disease to arresting and potentially repairing the damage done by the disease; in the case of the MRF repairing the damaged myelin sheathing around our nerves caused by MS.

While this may look like a retreat from the “real” goal, it should be seen as a necessary confirmation strategy in the face of significant novelty. By focusing on repair the MRF was choosing to stop or reverse the damage done by the disease (like AIDS) instead of having to address greater sources of novelty that would have to be identified and their value determined to cure the disease. With repair you only have to address the sources of novelty associated with repairing the damage done to the body’s capacity to produce myelin (so remyelination or myelin repair). Johnson clearly recognized the illusive nature of a cure focus as he was still waiting for one for over 25 years. Repair as a fundamental organizing principle for the MRF not only limits the sources and amount of novelty to be addressed, but makes what is identified and worked on easier to share and confirm with others outside of one’s discipline. A repair approach then functions as an initial sweet spot that makes it easier to address the challenges of high novelty in a well-researched but still poorly understood disease.

With this repair focus made clear, Johnson and the MRF’s Chief Operating Officer, Rusty Bromley, second move was to select a multi-disciplinary set of scientists, Principal Investigators (PIs), who would be primarily responsible for identifying and generating novel approaches as well as confirming its value. PI’s were selected based on their disciplinary focus, an interest in remyelination and a demonstrated interest in collaborating across disciplines. Confronting the challenge of collaboration among laboratories and their scientists was a significant change from the status quo. Bromley reflected on the collaboration challenge in academia: “most research laboratories work under an extreme regime of secrecy, as scientists are typically rewarded on the basis of priority--being first to publish. This pressure obviously militated against sharing.” Thus, typically, only after several years of data development, writing that up, and publishing it in a peer-

reviewed journal would scientists be willing to reveal the true focus of their work and share with others what they discovered. And further what would be revealed was what worked rather than what didn't and why. Bromley added: "so what we at the MRF demanded of our PIs was the discarding of academic norms so as to share the results of what would *and would not work*--with other laboratories, as it all happened." For the PIs who became involved the MRF insisted on new work arrangements among the scientists and their laboratories so that knowledge about remyelination could flow rapidly and freely between them.

To start this selection Johnson compiled a list of some 125 academic scientists with research in the myelin arena that could potentially join MRF. He refined the list based on those who were interested in repair, were highly productive, were full professors, were specialized in different aspects of MS and who had a reputation for collaboration. Based on these criteria and advice from scientists he had gotten to know, Johnson eventually narrowed down the list to five high profile scientists⁷ from five different institutions (Case Western University, University of Chicago, McGill University,⁸ Northwestern University and Stanford University) that represented each of the key scientific specializations that were implicated in creating a potential treatment for myelin repair. He planned to raise approximately 50% of each researcher's laboratory funding budget in return for securing their time and effort towards a novel collaborative arrangement.

When the scientists agreed with to work jointly on problems, their first task as a team was to list the top 100 scientific questions that needed to be answered for the development of a viable repair treatment for MS. Over two days, the PIs, along with key facilitation by Johnson and Bromley, brought to bear what they knew and didn't know about myelin and MS and then collectively agreed to the key research questions that would guide their scientific activities around remyelination. As these issues were identified the PI's began to realize that despite differences in disciplinary background and language that much of what they wished they knew focused on similar problem areas and key mechanisms. So from those 100 questions, six problem areas were identified as key gaps that the MRF would work on which became their integrated research plan (Figure 5). What made this third move of constructing these shared problem areas of such value is that they also defined key dependencies between disciplines and the scientific work to be done. This brought a clear focus to the projects would drive the scientific work to be done across

⁷ A typical researcher had a laboratory with 13 to 16 post-doctoral fellows, graduate students, and technicians.

⁸ The PI from McGill University left the MRF in 2008 due to increased administrative duties at his home institution.

disciplines and labs; which set of PIs would be involved; and even more specifically which “repair” hypotheses would be the starting points (see Figure 5 for these details). What we see here is that the shared sweet spot of a repair approach become populated with sequences of shared scientific activities where novel hypotheses could be generated and their consequences or value confirmed (or disconfirmed as the case may be) and next steps could be rapidly taken.

<insert Figure 5>

Now that the PIs had begun to work on the joint development of “repair” hypotheses, the PIs and the MRF staffs fourth move was to focus on the development and use of specific tools to develop novel approaches and confirm value. Such artifacts need to have a capacity to represent and test the dependencies of what is novel and what is known. The consequences generated from this helped actors determine what knowledge to transform and develop the next course of action that would best lead to developing a “repair” treatment. In these efforts we see the specifics of the infrastructure under development; specific shared practices and artifacts used to work together (i.e., repair hypotheses, testing standards, databases, mouse models, etc.).

Mouse models are an example of artifacts that provide significant value in scientific research (Murray and Stern, 2007). Mouse model are used to test particular treatments or isolate a specific biological mechanism or gene associated with a particular disease. Given the lack of knowledge about MS and its multi-causal nature isolating some or any of the mechanisms associated with MS would be a big breakthrough. In 2009 MRF PI Brian Popko’s team at the University of Chicago developed what they called a demyelinating mouse. What is critical here is that Popko’s team identified a gene that when not present results in a mouse that can’t remyelinate despite the present of healthy proteins used in remyelination. This mouse model is of particular value in addressing the novelty associated with MS because it avoids many confounding “dependencies” associated with such a multi-causal disease (genetic, environmental, autoimmune, viral, etc.) that would require testing to confirm their value. Since the MRF focus is on remyelination such a target mouse model allows a narrowing down of the potential novelty that would have to be addressed as well as a highly confirmatory artifact to test a variety of specific repair hypothesis and potential treatments. This mouse model provides a testing platform with a greater capacity to confirm the value of novel approaches and drive the N-C-T cycle associated repairing damaged myelin. Each of these moves builds on each other to establish an infrastructure

with greater capability to develop novel courses of action and confirm their value in this challenging scientific area.

5.1 An MRF Problem Solving Vignette: How the New Infrastructure Rapidly Develops Hypotheses about Repair

All the PIs describe how the MRF provides them with a new capability to rapidly confirm and disconfirm novel hypotheses that lead to more rapid scientific innovation. Examples of this increased capability for innovation in the MRF can be easily identified at any of the large face-to-face meetings with all the PI's, post-docs, scientific advisory board members (SAB) and MRF staff that happen four times a year. The first time we observed this was in October of 2006 in Montreal in a research presentation right after lunch. Ben Barres (PI from Stanford) was presenting his labs most recent set of experimental results. The presentation detailed the importance of a particular protein binding site that was thought to be beneficial in driving the re-myelination process. During the presentation a number of qualifying questions were asked by the audience about particular biological mechanisms as well as comments about findings from other studies that were thought to be consequential to the experiment being discussed. Near the end of the presentation Ben outlined the next hypothesis that this research would pursue based on the evidence generated from the current experiments. With this potential hypothesis now clearly stated a very energetic conversation began with all the PIs and a number of the SAB members joining. Subject matter of this conversation was evidence from a variety of studies; some published; some unpublished but recently reviewed. None of the studies referred to were familiar to all in the group, but between the highly diverse set of PIs and SAB members there was someone with the necessary expertise who could challenge, disconfirm or confirm the consequences of a given article, result, piece of data or Barres proposed hypothesis.

What the group started to build toward was the question if there was a better “next” hypothesis for Barres’s lab to focus on? Did the MRF now know more as a group to transform what was currently proposed and develop a more “innovative” repair approach? When the discussion ended Barres and the other PIs and SAB members had agreed that his proposed next hypothesis was no longer as consequential in developing a treatment to repair damaged myelin as initially thought. What was more surprising is that the group developed and then confirmed a “better” next repair hypothesis and they did so in only four minutes.

After the presentation we spoke to two of the PIs and three of the SAB members directly involved in the conversation we just observed. We posed the question: “Is this type of process of discovery unusual within the MRF?” The quick answer by all was no. We next informed them that the process of confirming the value (disconfirming) the proposed “next” hypothesis and then confirming a new one had only taken four minutes. They agreed that was a typical time frame for such discussions. The next question we asked was “How long this would have taken before the existence of the MRF?” The timetable given to us varied from four to five years if it would have happened at all. They talked about the long cycles of developing the science within an individual lab, assembling data and developing tools, then getting through the peer review process without revealing too much of the key science, and then the remote possibility of having such a detailed presentation and follow up conversations with such a broad but interested set of participants. We then made a symbolic contrast between “*four years to four minutes*” and all found this contrast shocking despite their agreement of the facts behind it. But they got over the surprise quickly, and one PI nodded and said, “I guess the more I think about it, this is often how things work; and this is how being a part of MRF makes our science faster and better than it was before.”

A faster innovation cycle is always a good thing, but the key question is how is such a cycle driven? The MRF infrastructure help actors generate more innovative courses of action by more rapidly developing novel hypothesis, confirming their value and based on the consequences identified determining what knowledge they were using to transform. All of this produces more innovation courses of action and potential treatments for repairing myelin.

6.0 Discussion: Characterizing N-C-T Cycles and Comparing Cases

To frame our discussion of these two cases we will characterize the OSS and MRF N-C-T cycles. Following that we will provide a much broader comparison across all the cases touch upon: OSS and proprietary software development (PSD), and the MRF and pharmaceutical and academic science (PAS).

Figure 6 is a summary of how the infrastructures of OSS and MRF differ in terms of a N-C-T cycle as the sweet spot of innovation. In the case of OSS given the nature of the infrastructure it has tremendous capacity to develop sources of novelty, but also significant capacity to confirm the value of that novelty. So it is effectively matching a strong capability for developing sources of novelty with a strong capability to assess the value of those sources. This is why the N-C-T cycle is much tighter and lower relative to the MRF. The similarly colored squares in the middle

of the sweet spot indicate the globally shared nature of the artifacts (i.e., boundary objects) present in OSS. The similarly colored circles represent the fact that even though the actors are distributed across different problem domains they have similar abilities to use shared artifacts as boundary objects upon which to share and assess their coding work. By contrast in the MRF there is not only more novelty associated with treating MS than fixing and developing software, but also the multi-disciplinary nature of the problems require both significant within domain specialization and cross domain collaboration. For this reason the N-C-T cycle is much larger and extends higher up than in OSS. We represent the greater specialization and variety of actors and artifacts with more multi-colored circles and squares. Not surprisingly these circumstances in the MRF result in a much longer N-C-T cycle as compared to OSS.

The artifacts used in OSS are the same as those used in proprietary software development (PSD), but are used quite differently. Not only do such globally shared artifacts such as a compiler and test suites create rapid cycles to determine the value or consequences of novel solutions, but further the artifacts associated with email communication and CVS allow actors to engage in the N-C-T cycle of code in a highly distributed manner. The infrastructure used in PSD follows a more controlled or industrial approach to product development (Cusumano, 1991) and so is less distributed and timely than in OSS. Much of this difference in distribution can be attributed to the unique intellectual property licensing schemes that are present in OSS communities. Open source licenses typically either mandate reciprocal sharing rights for software code (example the General Public License used in Linux) or have a free reuse terms (example the Berkeley Software Distribution (BSD) license used in Apache).

Now while some could suggest that such licenses are anti-intellectual property or anti-patent, they should be seen as an agreement among actors on how their work will be shared. As an artifact in the infrastructure the impact of such licenses on OSS communities is that it increases substantially the sources of novelty that can be generated and shared as well as increases the infrastructure's capability to determine the value of those increased sources of novel code. Further since a larger variety of actors can participate in OSS this rapid cycle infrastructure and their highly confirmatory outcomes makes valuing their abilities to assess and create code quite easy. The nature of the infrastructure is an ongoing means of selecting in skillful or "knowledgeable" actors into the infrastructure. This is why the infrastructure of OSS has tremendous capability to develop sources of novelty and determine their value to improve existing code.

However, it is important to point out that the value of such shared licenses as an artifact in software development would not have taken place without the rest of the infrastructure as described in OSS. If you didn't have a globally shared artifact like a compiler and test suites then there is very little innovative value in sharing the source code since there is not means to confirm its value. If you didn't have a globally distributed email then the costs of sharing and assessing source code go up astronomically. These licenses are an essential artifact in the OSS infrastructure that allows for a significant increase developing and determining the value of novelty, not just a stand alone story about who controls intellectual property. So any intellectual property stance, whether patent or licenses, should be seen as one of many artifacts in the infrastructures and as such its particular impact of the capability of that infrastructure to drive the N-C-T cycle must be considered, not just as a "thing" that defines who benefits from it economically.

Considering intellectual property as an artifact that impacts the distributed nature of the N-C-T cycle is a clear way to understand how and why the MRF differs from current for profit and academic approaches to drug discovery. Commercial drug development has a clear intellectual capital focus on a patent approach to drug discovery and the economic benefits that follow. Academic science primarily focuses on intellectual property as a part of the processes publishing, tenure and securing funding/grants. Although both these two approaches to intellectual property were stable for a long time the current productivity crisis in drug development has begun to question these approaches (Munos 2010). In the particular case of academic science despite the explosion of scientific knowledge as measured by the amount of published scientific papers that knowledge often remains within specialized domains driven by the requirements of tenure; so often lies unconnected from other knowledge domains about the human body that could lead to more effective treatments.

For commercial pharmaceutical companies most treatments developed have been for single causal diseases and so their siloed scientific efforts and patent focused approaches have worked (Pisano 2006). However, these knowledge and IP silos are proving problematic given that most remaining diseases are multi-causal and require multi-disciplinary efforts to develop potential treatments. Because of the specialization of knowledge within academic careers and disciplinary silos and the secrecy and narrow notions of IP that follow, the MRF had to develop a more distributed infrastructure in terms of knowledge and ownership given the multi-causal nature of MS. Current pharmaceutical and academic science (PAS) approaches is an infrastructure with

highly specialized actors and artifacts take years if not decades to develop. We see this specialization and long time frames in the educational systems of scientists were they start with an undergraduate, PhD, post doc (perhaps two) and then an assistant professor and eventually a tenured professor. And as our four years to four minutes story reveals it can take many months and often years to test the potential value of a novel hypothesis (i.e., a novel course of action). Given this existing infrastructure the MRF had to go through a number moves (a repair approach, selecting 5 PIs, identify dependencies across repair approaches, developing artifacts for day to day activities and real time information flow across experiments and labs) to develop a new infrastructure that could distribute the innovation task across more locales while at the same time accelerating the speed upon which the value of novelty was confirmed and innovative course of action developed.

In comparison to traditional approach to drug development the MRF should be seen as moving higher up in the novelty space not only because so little is known about MS, but also so much new infrastructure needs to be developed to pursue this across a variety of disciplines. This is why those involved in the MRF had to select particular actors and also develop artifacts to better address this multi-causal disease in a multi-disciplinary fashion. By comparison traditional pharmaceutical and academic science (PAS) approaches remains lower down in the space because they continue to focus on more traditional single causal approaches and their associated siloed orientation. The past successes of these PAS approaches also lead to a combination of tight controls over IP for ownership and of secrecy to insure academic status.

In Figure 7 we offer a rough comparison of proprietary software (PSD), OSS, pharmaceutical academic science (PAS) and the MRF along the dimension of the relative amount of novelty to address and the relative newness of the infrastructure required to address that novelty. We dimensionalize the amount of novelty from low to high and the status of the infrastructure as existing to new (needs to be developed). In the case of software, OSS is addressing similar problems to PS in term of the amount of novelty, it's just that they take a more distributed approach and so add the artifact of shared IP regime that allows them to develop and confirm more sources of novelty. In the case of drug discovery, the MRF is addressing the multi-disciplinary nature of the scientific challenge so are much higher up in the novelty space relative to PAS and because of that they also need to develop a lot of new infrastructure to be successful.

These comparisons are crude but do offer up a “relative” scaling of innovation problems and ways of organizing for them along the dimensions of the amount of novelty and newness of the infrastructure required as we naturally make comparison across empirical cases documented in the literature. Further we see that the N-C-T cycle helps us move beyond the current dualisms found in the literature as tension or combination of old and new (Galunic and Rodan, 1998), or ambidexterity (Tushman and O’Reilly, 1996) or even exploration and exploitation (March 1991). By specifying this cycle innovation helps ask a series of new and more specific set of empirical questions than before. For example, what is an adequate capacity to develop sources of novelty and how are new capacities developed in a given context? What abilities do actors need to develop sources of novelty and confirm their value? And how is this new way of organizing for innovation better than another in terms of addressing the challenge of novelty? Raising and then addressing these practical questions using the concrete and measurable categories of infrastructure gives us a clear avenue to describe, compare and begin predicting the relative performance of different forms of organizing for innovation.

7.0 Conclusion

We feel that studies of innovation are at a critical stage of development; of moving from an initial stage of developing descriptive theories of innovation to developing more formal theories of innovation (Glaser and Straus, 1967). Overall, to facilitate this movement we focused on the challenge of novelty, a problem that cuts across all contexts of innovation, and compared how it is addressed across a variety of circumstances (Christensen and Carlile 2009). To adequately understand the challenge of novelty we developed both a conceptual and empirical framework. Conceptually, the novelty-confirmation-transformation cycle describes how the challenge of novelty is addressed. Empirically an infrastructure for innovation specifies how actors develop and use artifacts to address the challenge of novelty. These frameworks were then applied to the PostgreSQL software community and the Myelin Repair scientific collaborative which allowed us to characterize the nature of sweet spot of innovation based on these very different circumstances. Then by comparing how an open source approach differs from propriety software development and how a more open approach to science differs from pharmaceutical academic science in terms of the relative amount of novelty and the relative amount of new infrastructure needed to address the potential novelty present provided a means of more formally comparing and theorizing across different circumstances of innovation.

References

- Afuah, A. N. 1998. *Innovation Management: Strategies, Implementation, and Profits*. Oxford University Press, London, U.K.
- Baldwin, C.Y. and von Hippel, E. 2009. Modeling a Paradigm Shift: From Producer Innovation to User and Open Collaborative Innovation, *Harvard Business School Finance Working Paper* No. 10-038.
- Black, L. and Carlile, P.R. "A Relational Capability for Knowledge Integration: Identifying Latent Problems in New Product Development Reviews." Boston University Working Paper.
- Bowker, G. and Star, L. 1999. *Sorting things out: Classification and its consequences*. MIT Press: Cambridge, MA.
- Business Week. 2009. Mega mergers can't cure the pharmaceutical industry. *Business Week*, July 26, 2009.
- Campbell, E.G., et al. 2002. Data withholding in academic genetics: Evidence from a national survey. *JAMA-Journal of the American Medical Association* 287, 473-480.
- Carlile, P.R. 2004. "Transferring, Translating and Transforming: An Integrative Framework for Managing Knowledge across Boundaries." *Organization Science*, 15, 555-568.
- Carlile, P.R. 2002. "A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development." *Organization Science*, 13, 442-455.
- Chandler, A. 1962. *Strategy and Structure*. MIT Press: Cambridge, MA.
- Christensen, C.M. 1997. *The innovator's dilemma: When new technologies cause great firms to fail*. Harvard Business School Press: Cambridge, MA.
- Christensen, C. M and Carlile, P.R. 2009. "Course Research: Using the Case Method to Build and Teach Management Theory." *Academy of Management Education and Learning*, 8, 2, 240-251.
- Cockburn, I. 2006. Is the pharmaceutical industry in a productivity crisis? *Innovation Policy and the Economy*, Volume 7. A. Jaffe, J. Lerner, and S. Stern (eds.). MIT Press: Cambridge, MA.
- Cusumano, M. 1991. *Japan's software factories: A challenge to U.S. Management*. Oxford Press: Oxford, UK.
- DiMasi, J.A., Hansen, R.W. and Grabowski, H.G. 2003. The Price of Innovation: New Estimates of Drug Development Costs, *Journal of Health Economics*, 22, 151-185.
- Dougherty, D. 1992. Interpretive Barriers to Successful product innovation in large firms. *Organization Science*, 3, 2, 179-202.

Dougherty, D. 2007. Trapped in the 20th Century? Why Models of Organizational Learning, Knowledge and Capabilities Do Not Fit Bio-pharmaceuticals, and What to Do about That. *Management Learning*, 38, 3, 265-270.

Downs, G. & L. Mohr., 1979. Toward a theory of innovation. *Administration and Society*, 10, 4, 379-408.

Drews, J. 2003. Strategic trends in the drug industry. *Drug Discovery Today* 8 (9).

Epstein, R. 2006. *Overdose: How Excessive Government Regulation Stifles Pharmaceutical Innovation*. New Haven: Yale University Press.

Food and Drug Administration. 2004. Innovation or stagflation: Challenge or opportunity on the path to new medical products. FDA White Paper.

Fujimoto, T. 1988. *The evolution of a manufacturing system at Toyota*. Oxford University Press.

Gal, U., Yoo, Y. and Boland, R.J. 2004. The Dynamics of Boundary Objects, Social Infrastructures and Social Identities. *Sprouts: Working papers on Information Environments, Systems and Organizations*, Vol. 4, Issue, 4, Article, 11.

Galunic, D.C. and Rodan, S. 1998. Resource Recombinations in the Firm: Knowledge Structures and the Potential for Schumpeterian Innovation. *Strategic Management Journal*, 19, 12, 1193-1201.

Glaser, B. and Strauss, A. 1967. *The discovery of grounded theory: Strategies of qualitative research*. London: Wledenfeld and Nicholson.

Gulley, N. and Lakhani, K.R. 2010. The Determinants of Individual Performance and Collective Value in Private-Collective Software Innovation. *Harvard Business School Technology & Operations Mgt. Unit Working Paper* No. 10-065.

Henderson, R. and Cockburn, I. 1994. Measuring Competence? Exploring Firm Effects in Pharmaceutical Research, *Strategic Management Journal*, 15 (S1), 63-85.

Honig, P. and Lalonde, R. 2010. The Economics of Drug Development: A Grim Reality and a Role for Clinical Pharmacology. *Clinical Pharmacology & Therapeutics* 87, 247-251.

Jantsch, E. 1980. *The Self-Organizing Universe*. Oxford: Pergamon Press.

Krishnamurthy, S. 2002. Cave or Community?: An Empirical Examination of 100 Mature Open Source Projects. *First Monday*, Special Issue #2: Open Source — 3 October 2005
<http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/viewArticle/1477/1392>

- Lakhani, K. and Wolf, R. 2003. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects . In: Feller, J., B. Fitzgerald, S. Hissam, K. Lakhani (eds.), *Perspectives on Free and Open Source Software*, MIT Press, Cambridge.
- Leonard-Barton, D. 1992. Core capabilities and core rigidities: A paradox in managing new product development. *Strategic Management Journal*, 13, 111-126.
- MacCormack, A., Rusnak, J. and Baldwin, C. 2006. Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code. *Management Science*.
- March J. 1991. Exploration and exploitation in organizational learning. *Organization Science*, 2, 1, 70-87.
- Munos, B. 2009. Lessons from 60 years of pharmaceutical research. *Nature*, 8, 959-969.
- Murray, F. and Stern, S. 2007. Do formal intellectual property rights hinder the free flow of scientific knowledge?: An empirical test of the anti-commons hypothesis *Journal of Economic Behavior & Organization*, 63, 4, 648-687.
- Murray, F. and O'Mahony, S. 2007. Exploring the Foundations of Cumulative Innovation: Implications for Organization Science. *Organization Science*, 18, 6, 1006-1021.
- O'Mahony, S. 2003. Guarding the Commons: How Community Managed Software Projects Protect Their Work," *Research Policy*, 16,15, 1-20.
- Porter, M. 1985. Competitive advantage.
- Ridley, M. 1993. *The red queen: Sex and the evolution of human nature*. Penguin Books: London, UK:
- Roberts, J. Hann, I. and Slaughter, S. 2006. Understanding the motivations, participation and performance of Open Source Software Developers: A longitudinal study of the Apache Project. *Management Science*, 52, 7, 984-999.
- Schumpeter, J. A. 1932/2005. Development,' *Journal of Economic Literature*, XLIII, 104–116 (Entwicklung) [Translated by M. C. Becker and T. Knudsen].
- Schumpeter, J. 1942. *Capitalism, Socialism and Democracy*. Harper Brothers: New York: New York.
- Scotchmer. S. and Green, J. 1990. Novelty and disclosure in patent law. *RAND Journal of Economics*, 21, 1, 131-142.

- Shen, X.X., Tan, K.C., and Xie, M. 2000. An integrated approach to innovative product development using Kano's model and QFD", *European Journal of Innovation Management*, 3, 2, 91 – 99.
- Sorensen, O. and Stuart, T. 2000. Aging, Obsolescence and organizational innovation. *Administrative Science Quarterly*, 45, 1, 81-112.
- Spear, S. and Bowen, K. 1999. Decoding the DNA of the Toyota Production System. *Harvard Business Review*, Sept-Oct, 1-12.
- Star, S. L. 1989. The structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving. In M. Huhns and L. Gasser (eds.), *Readings in Distributed Artificial Intelligence*. Menlo Park, CA: Morgan Kaufman.
- Star, S.L. and Rehleder, K. 1996. Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces, *Information Systems Research*, 7, 1, 111-134.
- Stern, J. & Simes, R. 1997. Publication bias: Evidence of delayed publication in a cohort study of clinical research. *British Medical Journal* 315.
- Teece, D. J., G. Pisano and A. Shuen. 1997. Dynamic capability and strategic management, *Strategic Management Journal*, 18, 509-533.
- Tushman, M. and O'Reilly, C. 1996. Ambidextrous organizations: Managing evolutionary and revolutionary change. *California Management Review*, 38, 4, 8-30.
- Van de Ven, A. 1986. Central Problems in the management of innovation. *Management Science*, 32, 5, 590-604.
- von Hippel, E. 1988. *The Sources of Innovation*. London: Oxford Press.
- von Hippel, E. & von Krogh, G., 2003, Open Source Software and the Private-Collective Innovation Model: Issues for Organization Science. *Organization Science*, 14, 2, 209-224.
- von Krogh G., Spaeth, S. and Lakhani K.R. 2003. Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, 32, 7, 1217-1241.
- Weber, S. 2004. *The success of open source*. Harvard University Press: Cambridge, MA.
- Winter, S. 2006. The logic of appropriability: From Schumpeter to Arrow to Teece. *Research Policy*, 35, 8, 1100-1106.
- Young, M.S. 1992. A Framework for Successful Adoption and Performance of Japanese Manufacturing Practices in the United States, *The Academy of Management Review*, 17, 4, 677-700.

Zollo, M., & S. Winter 2002. Deliberate Learning and the Evolution of Dynamic Capabilities. *Organization Science*, 13, 3, 339-351.

Figure 1: The conceptual sweet spot of innovation

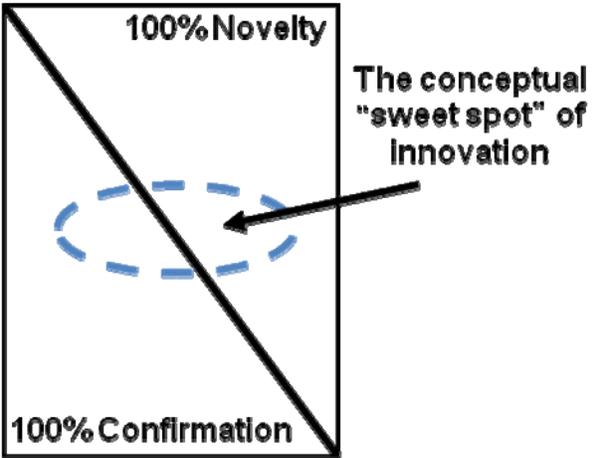


Figure 2: Establishing a sweet spot of innovation where the Novelty-Confirmation-Transformation Cycle can occur

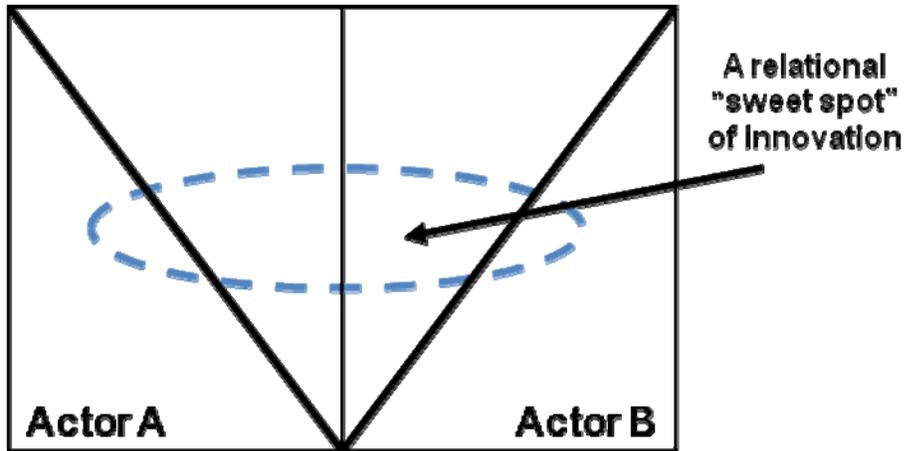
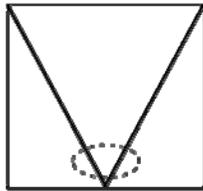


Figure 3: Different circumstances of the Novelty-Confirmation-Transformation Cycle

Open Source Software



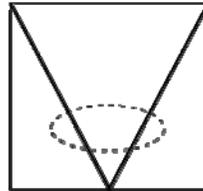
Artifacts

- "0s and 1s"/code
- compiler
- CVS repositories
- e-mail lists

Actors

- users=programmers
- core and periphery
- teaching/learning

Toyota Production



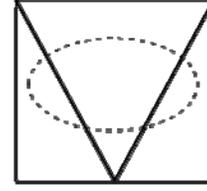
Artifacts

- parts/cars
- parts bin/kanban card
- tools
- production line

Actors

- production workers
- team leaders
- trainees/coaches

Myelin Repair Found.



Artifacts

- scientific data/hypoth.
- mice/samples
- presentations
- journal articles

Actors

- founders/entrepreneurs
- academic scientists
- post docs
- advisory board
- MS patients

Figure 4: Cycles of Innovation is OSS

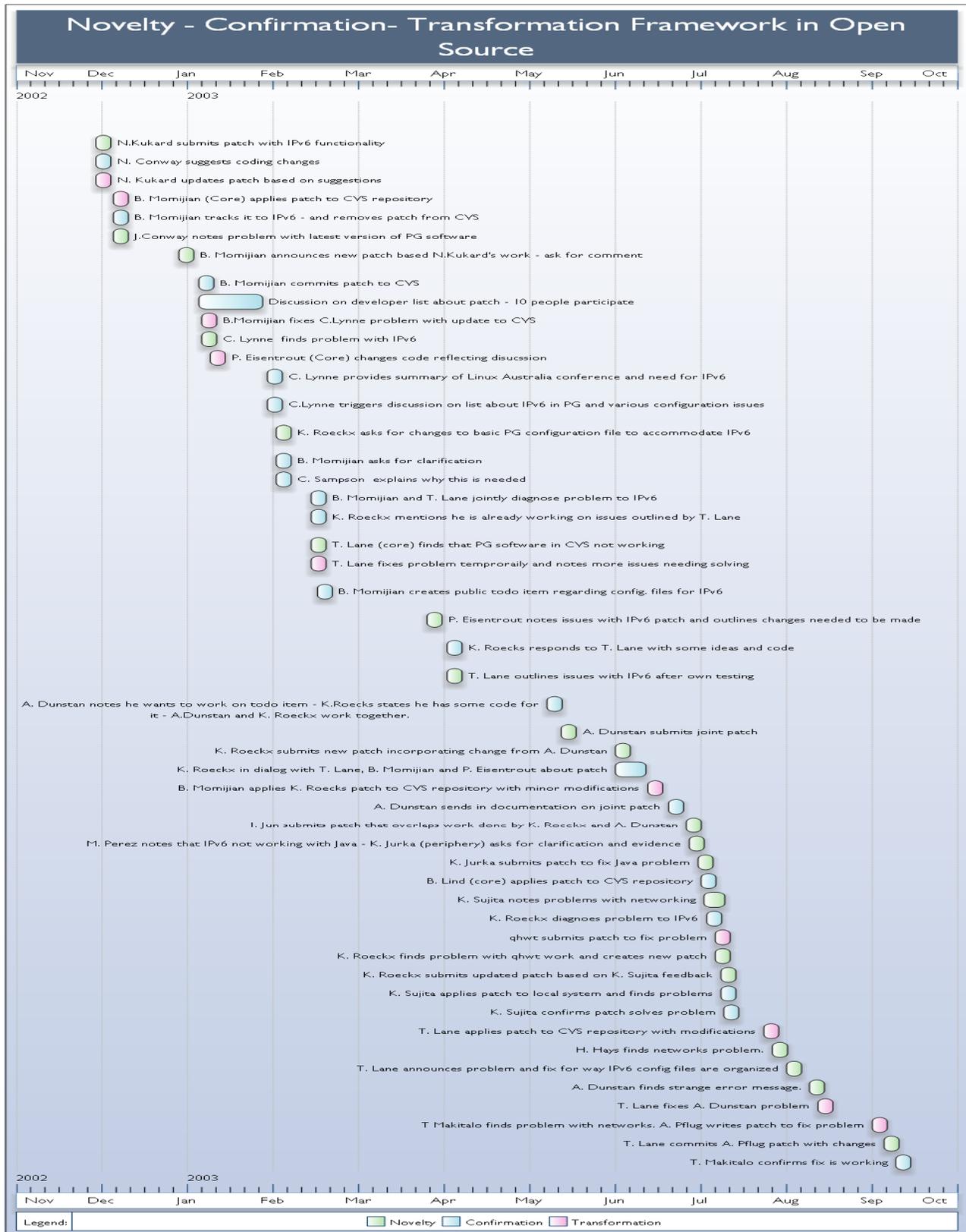
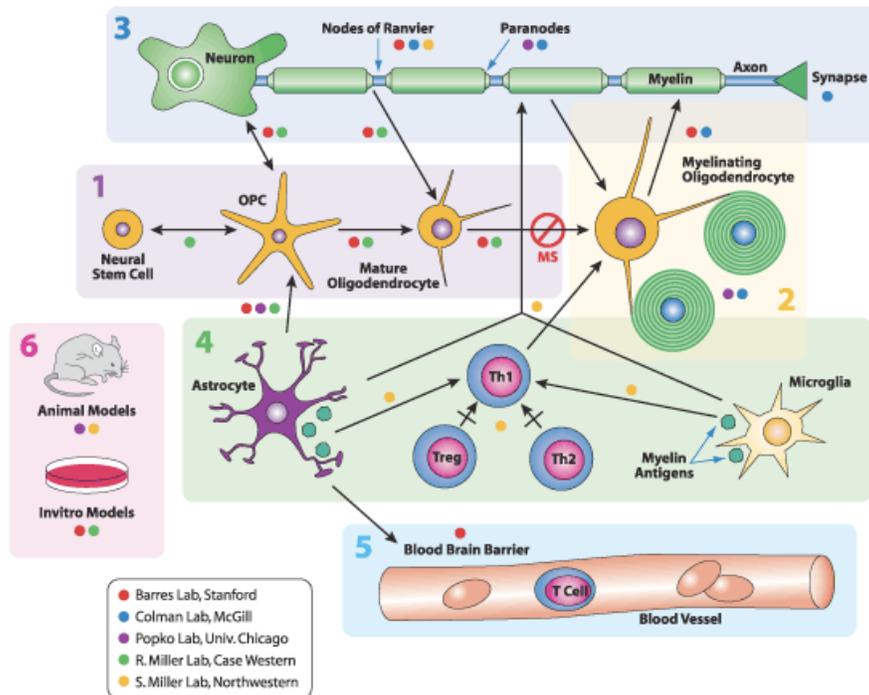


Figure 5: Identifying the Dependencies across different Disciplines at MRF



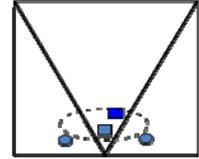
Research projects in these six key areas of investigation are focused on identifying novel therapeutic targets to arrest the MS process and/or promote remyelination:

1. Understanding how oligodendrocytes are normally generated from neural stem cells and how MS perturbs this process.
2. Understanding the underlying mechanism of myelination and how it is perturbed in MS.
3. Understanding how nodes of Ranvier and paranodes are normally formed and how they are perturbed in MS.
4. Understanding the immune response in MS and how inflammation affects myelin repair.
5. Understanding how the Blood-Brain Barrier is effected in MS and its role in the disease.
6. Development of better animal models for study of MS and remyelination

Figure 6: Characterizing OSS and MRFs N-C-T Cycles

□ **OSS's infrastructure for innovation**

- **Capacity of artifacts:** Artifacts are globally shared with well specified dependencies so makes for shorter N-C-T cycles
 - Makes changes to software easier to develop and evaluate
- **Ability of actors:** These artifacts make it easy to select knowledgeable actors
 - Actors get rapid feedback and the 'value' of their work can be easily assessed
- This "existing" infrastructure drives the N-C-T cycle more rapidly than was possible before in software development



□ **MRF's infrastructure for innovation**

- **Capacity of artifacts:** Specialized, numerous, overlapping and conflicting make for longer N-C-T cycles
 - The move from 'cure to repair hypotheses' shortens, increases the speed' the N-C-T cycle
- **Ability of actors:** The development of this new infrastructure can only be done by highly experienced actors
 - Actors require many years of experience to use, develop and make trade offs across artifacts
- This new infrastructure drives the N-C-T cycle more rapidly than was possible before in MS drug discovery

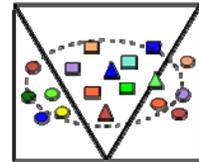


Figure 7: Comparing across Cases: Proprietary Software Development (PSD), Open Source Software (OSS), Pharmaceutical and Academic Science (PAS), Myelin Repair Foundation (MRF)

