2009

Reluctant Software Developers - Tactical Software Management Issues for Developing Software Outside the Software Industry

Lorie Obal
Claremont Graduate University, lorie.obal@cgu.edu

Reluctant Software Developers – Tactical Software Management Issues for Developing Software Outside the Software Industry

Lorie Obal
Claremont Graduate University
lorie.obal@cgu.edu

ABSTRACT

The issue of strategic software development management generally assumes the organization is in the software industry. However, there are cases of non-software vendors who have successfully built from within and even disseminated the results within their industry. These organizations must also grapple with strategic and tactical software management issues. Case studies and comparisons of such organizations may provide information for software organizations as well as unique perspectives on tactical management issues. Two cases, one a private sector company and the other a government agency will be examined to illuminate issues for building from within or insourcing. A government created, public architecture for Open Source Electronic Healthcare Records (EHR) systems is examined for potential leveraged development through an SPL approach.

Keywords

Tactical Software Management, Open Source, Healthcare, Software Product Lines

INTRODUCTION

Tactical software development management is not always restricted to software companies. Organizations with sufficient domain expertise may opt for a build from within or insourcing solution to leverage organizational competencies. Such cases might provide additional insight into software management due to their different organizational missions as compared to software vendors. The potential to study issues such as Software Product Lines (SPLs) in these organizations may be hampered by lack of access. Public government projects are an opportunity to view these processes with greater transparency. Such organizations can then be compared and contrasted with available case studies performed in the private sector.

An SPL is defined as " a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and are developed from a common set of core assets in a prescribed way" [21]. Key concepts include a core assets base and production of related products [21]. A focal point is a shared architecture as the foundation [3, 21]. Reuse in this context is defined as strategically planned rather than opportunistic reuse of components [21].

CASE STUDIES FOR COMPARISON AND CONTRAST – BTC BANKING AND THE VHA

One example is a build from within solution that was developed in the banking industry. The organization, alias "BTC Banking", developed expertise in banking service applications illustrating elements of Software Product Line (SPL) methods and software reuse [30]. As a part of a well-developed business strategy, BTC developed a sophisticated set of applications that could be easily configured and used with minimal or no programming effort to meet the diverse banking needs of its branches in different countries. Witman, (2007) coined the concept of "large scale software reuse" to refer to such practices within organizations. The software innovation manifested a solution to common banking needs and was scalable and customizable enough that it probably could have been commercialized, for example, through a spin-off and sold to other banks. A key feature of this innovation was the immersion in and understanding of the problem domain. However, the bank did not build on this opportunity because it did not want to diversify into the software business.

A case for comparison and contrast is the Electronic Healthcare Record (EHR) system VistA/CPRS developed by the Veterans Health Administration (VHA). VistA is the enterprise level architecture and CPRS is the clinical interface. In reference to the SPL approach, the VistA architecture forms the core assets and permits sharing of information within an

integrated patient record supported by different clinical specialties. Building upon the core architecture, new modules can be added to address specific healthcare information goals such as managing information for new clinical needs. For example, the original conception for VistA was to meet the healthcare needs of US Veterans. The VistA architecture permits adding new modules; therefore modules supporting the special requirements for treating pediatric patients can be added when adapting VistA for use outside the VHA.

The VHA is a significant case to study due to current interest in developing a lagging healthcare IT infrastructure. As in the banking example, VHA programmers were able to leverage their immersion in the clinical environment (understanding of the domain) and develop software that works with the fluid and often changing nature of clinical workflows [17]. This factor has made EHR development problematic for outside developers and is a recurring theme used to describe healthcare IT failures [24].The extent of immersion included the collaborative efforts of between 5,000 to 10,000 physicians and nurses actively involved in programming [19]. This included a pioneering use of agile development techniques to capture immediate feedback from clinical Subject Matter Experts (SMEs). However, the VHA may be more extreme in particulars of development and management partly because (1) the EHR system was not originally built as a part of a well-developed business strategy of VHA and (2) it has been made available to other healthcare organizations beyond VHA through the Freedom of Information Act (FOIA).

The VHA has the largest Electronic Health Record (EHR) system in the US. However, the VHA did not want to build the software which was initially built in secret by employees [25]. Eventually, the success of the system was acknowledged and accepted by the VHA and the system is now regarded as being an instrumental component in VHA quality of healthcare delivery improvements [2, 16, 18, 25]. This system is the interoperable backbone of the VHA health records system and is in place at over 155 VHA hospitals and clinics. Due to the scope of the system, it has been suggested as a national level EHR [4, 22] and as such, presents the potential for studying large scale software reuse [30] and management. Recent Presidential support for a large-scale shared health record system between the Department of Defense (DoD) and the VHA [12] is setting the stage for a new large scale health record architecture. The participation of the VHA is significant because it has the largest, open source electronic healthcare record (EHR) system in the US. This is an opportunity for strategic development and leverage of existing EHR core assets using the principles of SPL development for use throughout the general population.

As in the banking example, the VHA management has not been inclined to diversify into the software business. The success of VistA in other organizations has surprised the VHA with different types of innovations including downward scalability to minimal hardware environments [31]. Public availability potentially allows support for re-use to come from outside, even to the extent of creating new markets that support reuse. For example, there is now a certification mechanism [6] for EHRs. This nonprofit agency has certified an Open Source version of VistA, WorldVistA EHR VOE/1.0 on criteria including functionality, interoperability and security as well as one of the variants of VistA called RPMS [6, 15]. Certification could be part of a management method to ensure consistency, robustness and standardization. An Open Source pre-certification, interoperability testing framework for EHRs [23] is currently available [7]. These resources can potentially allow the organization to use the certification process as a form of outsourced assessment.

INNOVATION MANAGEMENT ISSUES

Both of these examples illustrate salient issues for comparison and contrast in tactical software development management. Potential problems for strategic management of the VHA model include attempts to privatize portions of VistA by incorporating proprietary software modules into the main VistA architecture, organizational conflicts and restructuring [19]. The VHA development method may be categorized with the insourcing model proposed by Hirchheim and Lacity (2000) as type 4. This is defined as senior executives confirming the value of insourcing even if there is not necessarily a cost savings, but it is still recognized by management as a success [14].

The VHA example is unique as it is the largest, longest running example of an EHR architecture. It has a mature core architecture, VistA, that is a key feature for the SPL approach [3, 21]. In addition, the coherence of its "product lines" i.e. – healthcare support modules exist in the original system as well as Open Source innovations outside the VHA. This raises the question: is it possible to also use the SPL approach from the Banking case study to manage future development for large government insourcing processes? This is a worthwhile question, considering that VistA/CPRS, and other non-classified systems developed by the US government are potentially available to a large number of outside stakeholders.

Insulation from the forces of the software industry is a common factor in both cases. The banking entity made the conscious decision to avoid those influences. The fact that the VHA is a government agency shields it from market forces that can otherwise shake out managerial inefficiency and competition for product superiority. Currently, a release governance program handles management of innovation/development at the VHA. Class III software is developed at local facility level and can eventually be approved for national level (Class I) implementation [27]. Public availability makes it possible to

subject VistA/CPRS to a different set of tactical management strategies including management methods of Open Source Projects. There are advantages to this for stakeholders both within and outside the government. Advantages for government include:

1. Receiving additional development at no additional cost
2. Possibly learning new work practices from industry
3. Additional innovation from entities that also have significant domain expertise

Advantages for Open Source developers include profiting from:

1. Creating training tools
2. Installation
3. Customization
4. Support

The latter group particularly benefits from the SPL approach since the business model is built upon existing core software. These principles have already been applied to VistA to a moderate extent. There are certified vendors who support VistA/CPRS installations outside the VHA such as Medsphere[1]. For example, VistA/CPRS has been notoriously difficult to install. This was not an issue for a government agency not necessarily concerned with support costs. Both the Open Source community and Open Source support vendors have created RedHat Package Managers (RPMs) that significantly facilitate the installation process of VistA/CPRS [20, 26]. Historically, the VHA has governed the VistA core assets. VHA policy governing VistA development has prevented project forking within the large VHA system, thus keeping the core architecture stable for adding new modules to support new processes. This is advantageous to Open Source developers building from VistA with an SPL approach as long as they can rely on the VHA for code management and assuring the consistency of the core assets.

The VHA has also determined variability management based on internal needs. This picture becomes more complicated since outside organizations will have to determine which parts of VistA they need and pre-existing dependencies between modules. This will involve an analysis of feature management including recognition of common features that must always be supported, optional features and alternative features [10].The possibility of open source community involvement generates the following research questions:

1. How can the SPL approach address Open Source development issues such as module dependencies?
2. How will architecture requirements for SPL development be defined and function under unique collaboration conditions that may include the flexibility inherent in agile development?

A comparison of SPL concepts to issues identified in the Banking case study and VistA/CPRS development was made to generate more specificity on issues raised by the research questions (See Table 1).

[1] http://www.medsphere.com/

Software Reuse Comparison	**BTC – Banking**	**VistA/CPRS**
1. SPL Concepts • Architecture – core asset base • Related products	• Java Banking Toolkit (JBT) [29] • Automated Teller Systems, Internet Banking, Worldwide Single Signon,	• VistA architecture –open [28] • Healthcare applications
2. Development Transparency	Closed – require access to company	• Open Source through FOIA • Public audits
3. Factors Contributing to Reuse Program • Technical • Organizational • Market • Scope	• Modification of business rules • Demonstrated large scope for reuse	• Certification (CCHIT) • Vendor Support • Potential for large scale distributed development
4. Leveraging change & Innovation	• Centralization of technology activity – minimizes barriers to reuse	• Potential to leverage open source community models • OpenVistA, WorldVistA • Government-private collaboration [11]
5. Organizational Characteristics	• Geographically distributed with regional units • Peer to regional organization	• Geographically distributed with regional units • Open development organization
6. Threats to this model	• Outside market forces	• Insertion of proprietary modules • Organizational restructuring

Table 1: Tactical Software Reuse Issues between VHA and BTC Banking Case Study

Potential Solutions and Future Research Focus

Table rows one and two reflect both the strengths of open source in SPL as well as one of the more perplexing qualities of VistA development. In both cases, the core architecture serves to support future development based on core assets. However, the demanding needs of the healthcare environment have historically fostered an emphasis on adaptive, agile development – used at the VHA before the term "agile programming" had been coined - to capture the needs of medical SMEs [25] [19]. This makes it difficult to retroactively classify VHA development along the continuum of model vs. code prioritization in the development strategy [1]. This issue is further complicated by the apparent use of a build as needed strategy at the VHA that can be seen in review of the VistA Monograph [28]. For example, VistA supports non-healthcare functions such as facility maintenance functions peculiar to VHA campuses. The promotion of a solid SPL approach requires additional clarification and codifying of operational development methods to facilitate standardization of SPL elements and effective communication

between internal and external developers. A potential solution for the necessary analysis may be possible due to the VHA's required transparency through public audits and FOIA.

Agerfalk et al. (2006) pointed out problems inherent in managing open source components for SPL use including: development including the lack of assigned roles for development and verification of models, responsibility for change control and development when only source code is available [1]. Potential solutions given in row three include certification and outside vendor support. Although the VHA has maintained a governance policy for developing and maintaining VistA, there is some indication that VHA management does not want the business of continually maintaining the code. Therefore, project forking remains a threat to VistA as a basis for ongoing development using the SPL approach. This would also be a major threat to Open Source developers seeking to build upon VistA in SPL fashion since they might no longer be guaranteed a consistent architecture as well as for organizations providing VistA to outside adopters using the Open Source customization and support business model.

One of the most exciting aspects of VistA/CPRS as a model for an SPL is the potential for innovation utilizing the resources of a large government agency in conjunction with outside innovation. Through the continued use of open standards, the core architecture could be spread among more diverse development groups and promote innovation driven by the introduction of new user needs [1] (See Table 1, row 4). To a limited extent, government-private collaborations are being investigated [11] and could leverage the VistA framework for robust, distributed innovations. As row five of table one illustrates, geographical distribution did not inhibit development due to the presence of reference architecture.

Currently there is a managerial threat to VistA/CPRS in the form of changed government policies. The US Congress is attempting to force the VHA to change its collaborative, decentralized development methodology into a centralized IT development model, which is currently alien to the prevailing culture of innovation at the VHA [19]. In addition, the sheer size of the VHA and its layers of political governance pose significant barriers to effective management of innovation. This makes the future path for tactical management and reuse uncertain. Conversely, the presence of stable reference architecture could alleviate the disruptive nature of organizational restructuring.

The most obvious solution for maintaining a coherent SPL core and managing variability would be for the VHA to be the keeper of the authoritative versions of the core architecture. Adoption by other healthcare agencies would entail a combination of feature analysis to determine functional re-use [10] as well as use case analysis to determine functional requirements [10] driving outside innovations and variability of modules derived from the original VistA core assets. This will be one of the most difficult issues to resolve, as it is dependent on changeable government policies.

Pressures toward proprietarization are a direct threat to the open source model advantages of VistA. The co-mingling of proprietary and public code can be subtle and creates legal barriers not only to standard open source distribution and can threaten an entire SPL foundation if it resides in a core asset [1]. The cascading effects of licensing schemes in this context are beyond the scope of this paper. However, proactive measures can be taken when reviewing SPL assets from the core and upward. The FOSSology project is an open source tool for analyzing software for the presence of proprietary licenses [13]. While this method may not be foolproof in the event of programmers using a copy and paste approach, it can be used to help protect the open integrity of core assets.

CONCLUSION

This paper has examined the concept of Software Product lines produced by entities that are not explicitly in the software industry. The use of open or public software systems may provide greater opportunities to examine tactical software management and software reuse. The efficacy of tactical management efforts in closed industries such as the banking example, are opaque to the outside observer. The current literature on SPL methods and tactical management tends to concentrate on proprietary industry examples [5, 8, 9]. This can be problematic for researchers lacking good industry contacts and access.

The case of VistA/CPRS presents a unique opportunity for SPL reuse methods with an architecture focus [3, 21]. In this case a public architecture creates the potential for outside collaborative development methods. Non-VHA health agencies will still have to analyze and manage feature variability as their needs may differ from the VHA's conception of variability management. It is not yet clear what will be the outcome of government policy to manage VistA development. The VHA is moving from its initial decentralized model, with innovation occurring at local clinics, to a centralized development and governance model. If the decision is made to hand it over either all in part to private vendors, the project may benefit from market forces, but the government may risk losing control over its own creation. Added risks include re-writing and closing the source code and vendor sun-setting. Another option is to work collaboratively with the Open Source community and

vendors. This option allows other healthcare organizations to adapt the work if they are not satisfied with the adaptations of a particular software vendor.

The literature on tactical software management and Software Product Lines has not leveraged the transparency of Open Source and public government software projects for research. In addition, the study of tactical management of software created by non-software vendors may provide additional counterpoint due to their divergent organizational goals. More work on an SPL approach may benefit governments and large healthcare agencies seeking to leverage their current EHR assets as well as facilitate EHR adaptation to continuing advances in medicine and changes in healthcare policy. The study of SPL approaches for EHR systems may prove useful not only for those seeking to leverage an existing architecture, but also for those charged with governing such projects. This is particularly timely considering current pressures to develop a national healthcare infrastructure in the US. Perhaps examining an SPL approach to EHR building before attempting national-level, public EHRs could result in cost savings through more efficient use of existing EHR assets.

REFERENCES

1. Agerfalk, P., Fitzgerald, B., Lings, B., Lundell, B., O'Brien, L. and Thiel, S., Open Source in the Software Product Line: An Inevitable Trajectory? First *International Workshop on Open Source Software and Software Product Line*, (Baltimore, Maryland, 2006), SEI CMU, 4.
2. Asch, S.M., McGlynn, E.A., Hogan, M.M., Hayward, R.A., Shekelle, P., Rubenstein, L., Keesey, J., Adams, J. and Kerr, E.A. Comparison of Quality of Care for Patients in the Veterans Health Administration and Patients in a National Sample. *Annals of Internal Medicine*, *141* (12). 938.
3. Bosch, J. Introducing Software Product Lines, University of Gronigen, 2001.
4. Brown, S.H., Lincoln, M.J., Groen, P.J. and Kolodner, R.M. VistA US Department of Veterans Affairs national-scale HIS. *International Journal of Medical Informatics*, *69*. 135-156.
5. Brownsword, L. and Clements, P. A Case Study in Successful Product Line Development *Software Product Line Case Studies*, Software Engineering Institute, Carnegie Mellon University, 1996, 98.
6. CCHIT. Certification Commission for Healthcare Information Technology, Certification Commission for Healthcare Information Technology, Chicago, Ill, 2008.
7. CCHIT. Project LAIKA, CCHIT, 2008, Open Source EHR testing framework to test pre-certification interoperability.
8. Clements, P. and Northrop, L. Salion, Inc.: A Software Product Line Case Study *Software Product Line Case Studies*, Software Engineering Institute, Carnegie Mellon University, 2002, 51.
9. Clements, P., Sholom, C., Donohoe, P. and Northrop, L. Control Channel Toolkit: A Software Product Line Case Study *Software Product Line Case Studies*, Software Engineering Institute, Carnegie Mellon University, 2001, 65.
10. Gomaa, H. and Olimpiew, E.M. Managing Variability in Reusable Requirement Models for Software Product Lines. In *High Confidence Software Reuse in Large Systems*, Springer, Berlin/Heidelberg, 2008, 182-185.
11. GOSCON. Government Open Source Conference, 2008, Annual conference to explore the use of Open Source software in the government sector.
12. HealthcareITnews. Obama announces new joint health record for DoD, VA. Monegain, B. ed. *Industry News*, MedTech Publishing, 2009.
13. Hewlett-Packard. FOSSology, Hewlett-Packard Development Company, L.P., 2009, Software tools to facilitate the study and analysis of Free and Open Source Software - used for finding software licenses in text.
14. Hirschheim, R. and Lacity, M. The Myths and Realities of Information Technology Insourcing. *Communications of the ACM*, *43* (2). 99.
15. IHS. Indian Health Service Electronic Health Records System Receives New Certification. Service, Indian Health Service. ed., 2009.
16. Kizer, K.W., Demakis, J.G. and Feussner, J.R. Reinventing VA Health Care: Systematizing Quality Improvement and Quality Innovation. *Medical Care*, *38* (6). I7-I16.
17. Lenz, R. and Kuhn, K.A. Towards a continuous evolution and adaptation of information systems in healthcare. *International journal of medical informatics*, *73* (1). 75.
18. Longman, P. *Best Care Anywhere: Why VA Health Care is Better Than Yours*. PoliPointPress, 2007.
19. Maduro, R. VistA & Open Healthcare News. Maduro, R. ed., Roger Maduro, Leesburg, 2008.
20. Medsphere. OpenVistA CIS, Medsphere Systems Corporation, 2008.
21. Northrop, L., Software Product Lines: Reuse that Makes Business Sense, *Australian Software Engineering Conference (ASWEC)*, (Sydney, Australia, 2006), Software Engineering Institute.

22. Oliver, A. The Veterans Health Administration: An American success story? *Milbank Quarterly*, *85* (1). 5-35.
23. ProjectLaika.org. Laika - An open source electronic health record testing framework, MITRE Corporation, 2009, An open source electronic health record (EHR) testing framework.
24. Silverstein, S. Why Healthcare IT Projects "Go Bad". Silverstein, S. ed. *Sociotechnologic Issues in Clinical Computing: Common Examples of Healthcare IT Difficulties*, Drexel University, Philadelphia, 2007.
25. Timson, G. The History of the Hardhats, Hardhats.org, 2008.
26. Valdes, I. WorldVistA RPM Installer. Valdes, I. ed., WorldVistA, 2009, Installs WorldVistA EHR/VOE 1.0 via rpm.
27. VHA. Modifications to VHA Class I Software. Affairs, Department of Veterans Affairs. ed., Veterans Health Administration, 2004.
28. VHA. Veterans Health Information Systems and Technology Architecture (VistA), United States Department of Veterans Affairs, Washington, D.C., 2008, Source page for the VistA monograph describing VistA functionality.
29. Witman, P.D., Software Product Lines and Configurable Product Bases in Business Applications - A Case from Financial Services. in *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, (2009), 1.
30. Witman, P.D. and Ryan, T., Innovation in Large-Grained Software Reuse: A Case from Banking. in *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, (2007), 228c.
31. WorldVistA. OLPC Health Project, WorldVistA.org, 2008.