XP Transition Roadmap

Emmet C. Eckman III
Denver Systems Operations
TRW Incorporated
750 South Richfield Street
Aurora, Colorado 80017 USA
+1 303 873 8751
Emmet.Eckman@auc.trw.com

ABSTRACT
This paper defines a roadmap for the transition of a large big design up front (BDUF) system from a waterfall development to an Extreme Programming (XP) Operations and Maintenance (O&M) phase. The roadmap addresses the political, management, technical and operational impacts to the transition from a traditional waterfall development to an XP development, as well as the change to the relationships with and between the customer, users and other related systems. Unless relevant, the roadmap does not address the impacts of transitioning from development into the O&M Phase. This roadmap is not designed in a vacuum – the target application is a C++, real-time, digital signal processing (DSP) system, over 2 million lines of code, running on a large cluster of Silicon Graphics connected machines designed and built by TRW Inc., Denver Systems Operations. The initial part of the roadmap anticipates and addresses the management and financial concerns of both program and customer management. The second part of the roadmap addresses technical concerns. The third part of the roadmap addresses the operational concerns. Finally, we address the political concerns the cut across the three previous parts of the roadmap.

Keywords
Extreme Programming (XP), XP transition, roadmap, plan, EVMS, BDUF and XP, EVMS

1 INTRODUCTION

The final phase of a classically developed BDUF system using the "waterfall model" is maintenance [1]. This phase includes both normal Operations as well as Maintenance (O&M) of the system. This does not mean that system development does not continue, rather, the opposite is true. While in O&M, defects are detected, reported and fixed as well as enhancements made to the system to increase functionality and improve performance. XP offers the opportunity to dramatically reduce the number of defects that propagate into the system from development. Detecting and fixing defects before they reach operations is critically important, as in this phase the cost associated with detection, reporting and correction is the highest. Each defect that escapes into the operational system is detected by one or more users, reported to a configuration control board (CCB), reviewed by the board members, scoped for effort to fix, analyzed for cost and schedule impact, prioritized, designed, fixed, tested and released in a new system baseline. The effort required to fix every defect reduces the available resources available to add new enhancements and capabilities. More than at any other time in the life cycle of the system, a defect in the operational system is most expensive to fix and costs more in overall resources (time and funding) when the system is in operational use. Consequently, the advantages to using XP are alluring. However, the concerns about transitioning to XP are daunting. This roadmap divides the concerns into four areas: management and financial, technical, operational; and political. Further, this roadmap examines the changes XP will bring to the relationships with and between the customer, users and other related systems.

Unfortunately, the program customer (who is not the user of the system) is not located in Denver, CO where the program development and maintenance occurs. In order to meet the geographic challenges, the customer has procured a System Engineering and Task Assistance (SETA) company in Denver to assist in contract oversight and maintenance. The SETA engineers are local to the development organization (they reside in the same building as the development organization). The SETA engineers will be the customer and user representatives during the iteration planning and will draft and manage XP stories. The SETA organization acts as a surrogate for the customer and users.

2 MANAGEMENT AND FINANCIAL CONCERNS

The principal management concern is managing change. Change management is difficult under the best circumstances; however, when that change cuts across all three areas of cost, schedule and performance, managing the effects of the change is made more difficult.

Management Concerns

The root of change in any organization is an advocate for the change. The advocate is the person who evangelizes, campaigns, champions, argues, challenges and fights for adoption of something new. To make the transition to XP, an advocate must be found (or created inside the organization). Preferably, someone who knows the system and has some legacy experience: someone who will be an advocate for XP. The advocate must have support to make the change (After all, it is impossible to program in pairs when there is only one programmer!)

We suggest an iterative approach to change management and build support for XP. Initially starting with management workshops engaging senior management and functional managers, leading through mid level

management workshops, including senior management participation, and finally ending with training for the SWEs, system engineers (SEs), architects and testers and trainers. The workshops provide an opportunity for buy-in to the concept of the adoption of XP as a methodology top-down (senior management to engineers). Further, the workshops allow for the opportunity for XP advocate(s) to address issues and concerns, and formulate policy and procedures early in the decision process to address those concerns.

The largest management concern is the shift of iteration and release planning from managers directly to the software engineers (SWEs) who do the work. It is not unlike the release of responsibility a parent makes when a child first learns how to drive. Using XP, SWEs, not managers, are responsible for determining what they can do (their velocity) and how to do it (their code). Manager's monitor and report progress.

The program customer requires the use of the Earned Value Management System (EVMS) [2] with Quantifiable Backup Data (QBD) for SW development. XP lends itself naturally to the adoption of the EVMS by using a single QBD template (implemented in Excel™ spreadsheets) for design, code and test for every task assigned to a SWE. Managers can quickly complete the QBD template from the information provided by each SWE at the daily meeting. Detailed metrics from the QBD, in addition to team velocity, are naturally collected and can be analyzed. Having these detailed metrics also alleviates the perception that XP is just 'hacking' and further addresses the management concern related to metrics collection and analysis supporting corporate SEI level 3 rating [3].

Iteration, release, deployment and update planning will initially be difficult, because velocities will not be known and the team will not fully understand how to use velocity and other metrics collected in the EVMS. However, planning will become easier as velocities and other EVMS metrics become better understood.

Financial Concerns

The principle financial concern is determining how many new capabilities can be procured and how many defects can be repaired on a fixed O&M budget. Unfortunately, TRW does not have a historical basis for estimating team velocity or the number of stories that can be procured by the program customer by dollar amount. Consequently, delay detailed financial planning until team velocity is better known and measured. Initial financial planning should be limited to the first few iterations. Most funding held in management reserve and allocated to development as velocity is better understood. Eventually, using XP, financial planning will be easier, and future forecasting conducted as the customer can approach planning knowing the cost of each capability or enhancement and an prioritize them.

Next, the cost of adopting XP needs to be addressed. XP requires the use of an integrated development environment (IDE), source code control system for configuration management (CM) and an automated test suite. The cost of change (adoption of a new IDE) needs to be understood, planned and phased over time. Know this cost before approaching the customer.

3 TECHNICAL CONCERNS

The primary technical concern is the adoption of a new IDE with integrated CM and automated test. The current program IDE does not have an integrated test suite, compilation times are slow, and the development environment is complex. The system has thousands of configuration files that tune the performance of the system. The turnkey change over to XP would be extremely disruptive and costly. The customer will not be able to tolerate all maintenance stopping for XP training, the IDE updates and process management. Consequently XP needs to be adopted in a phased approach. The program development teams are already naturally broken into three subsystems: interactive, automatic and infrastructure. The phased introduction of XP by subsystem is less disruptive and less costly that a turnkey solution. Further, it allows the development team to phase in training as well as work out process 'kinks' in small groups. Further, the phased introduction of XP helps manage the change process.

The roadmap advocates starting with the smallest subsystem first: the infrastructure subsystem. Starting with the smallest subsystem helps lower the risk to change while giving the team experience and hopefully increases the change of demonstrating success early. (By working with the system as a collection of subsystems, the development teams remain small and focused. Code ownership and understanding is at the team and subsystem level).

Because interactive tests are key to a successful XP development, and the program baseline does not have tests at this level, the initial iterations will be solely dedicated to stories for the development of automated tests. The new unit tests, which can be run interactively in the IDE, should be based on the current acceptance tests and test procedures. These stories should be written jointly with the test and SETA engineers. The initial iteration programming pairs consist of both development and maintenance SWEs as well as test engineers. In addition to the initial unit test stories, the program also needs to address long compilation (build) times. This should also be addressed as a story(ies) for the initial iteration(s).

When the automated unit tests are complete, then the team shifts to he introduction of new code (to support defect fixes or new capability development). Until all three subsystems are using XP and fully integrated, the old processes for design, development, integration and test will be used at the system level.

Our large DSP program has heuristic/Monte Carlo/parallel algorithms that do no lend themselves to strictly deterministic results for any realistic input data set. Further complicating the testing, is that performance testing must be conducted on the entire system. This necessitates the retention of a separate test organization for performance testing after unit testing. Conse-

quently, some stories are not complete until acceptance tests indicate all performance characteristics have been met. These tests cannot be automated.

In a large system, monitoring change to the architecture is traditionally the job of system engineers in the OOA/OOD or task analysis phase. Using XP, impacts to the architecture are monitored during iteration and task planning. System engineers and architects must participate in iteration planning and development. Where stories or tasks may impact the architecture, architects meet with the task owner and review design.. Decisions about design changes are documented and submitted to the Architecture Review Board (ARB). XP reduces the desire for up front system engineering. Finding the design which is good enough at the moment when it is required (80%), rather than the optimum design (simple is really better) is the key. That last 20% can easily double the cost [4].

4 OPERATIONAL CONCERNS

The program customer is concerned about protecting the operational baseline from unanticipated change and the propagation of defects (system operations) and the reproducibility of the development process over time (program operations). Our customer, like many others, is risk adverse.

System Operations

The essential operational concern is the successful operation of the system: put simply, the system is being used and is producing significant value for the customer. The primary allure of XP is the inherent integration of testing in the development process (early defect detection and correction) which helps prevent the introduction of defects into the operational baseline. XP helps the customer better manage change (risk) by enforcing rigorous testing early and often. This point must be emphasized to the customer and management.

Program Operations

In addition to the continued successful operation of the system, the customer is also concerned about the successful maintenance of the operational system. Historically, this concern has been expressed as the reproducibility of the development processes and production of system artifacts.

Currently, the ARB guards the architecture from unplanned and unexpected changes through participation in the up front design process. Our historical approach to failures in defect detection and correction is to add layers of review (new processes and boards) in order to manage the change to (*e.g.* guard) the operational baseline. The XP emphasis on test -- that is, build to test, rather than test what is built -- will not only reduce the number of defects which propagate from development to test to operations, but will also drive changes not only in the way acceptance testing is performed.

An effect of adopting XP is the significant reduction (but not the elimination) of the number of system engineers and architects to review design and staff review boards like the ARB. Because the system is very large and complex, no one person understands the entire system end-to-end. System engineers and architects will still b required, but when and where they perform their job will change over time.

5 POLITICAL CONERNS

Political changes generally overlap with management, financial, operation and technical concerns.

Staff Concerns

Some SWE staff will not be willing to migrate to paired programming and some managers will not accept the organizational changes from adopting XP. Functional management will have to help to identify staff that is unable or unwilling to make the change and resource them to different projects. Replacements will need to be culled from other projects or new hires.

Staffing changes will need to be addressed by program management as the need for up front system engineering is less and participative system and software engineering is needed more. Further, the shift of the integration process from a separate integration and test (I&T) organization to the development organization (as a natural part of XP developments) will decrease the need for integration and test engineers. As these subtle, but important shifts in staff naturally occur, a new organizational structure will evolve to meet the needs of the new functions which will occur as a part of XP.

Training

The XP development process requires training and mentoring. Starting with the test and QA organization, through the subsystem development organizations and concluding with the system engineering organization, initial training is required. Initial XP training, or XP immersions needs to occur before the first iteration occurs and should include the appropriate SETA and program management staff.

XP, Engineering Processes, SEI and CMMI

As a part of adopting XP, TRW should introduce a story into an early iteration to tailor standard TRW engineering practices for XP. Also TRW should introduce another story into an early iteration to compare ad contrast XP development methodologies against corporate TRW development practices to examine any changes or adaptations to our SEI or Capability Maturity Model (CMM®) Integration℠ (CMMI) certification [5]. We must make sure that our XP processes ‘fit’ our rating.

Documentation

Through the BDUF development, the customer has come to expect, and the developer has come to produce documentation at every phase or step of the process. The desire to document has increased as defect detection and correction has moved closer to the operational system and farther from development. Primarily, in hopes that a better documented system will reduce risk of unplanned change to the operational system and detect defects. Staffing reductions and turnover only increase the pressure to document the system for future use. We propose a very streamlined documentation process, especially at reviews and force more emphasis on the product (the code) than the process.

Implementing XP compacts the traditional steps of object oriented (OO) analysis (OOA), design (OOD), code (OOP) and unit test (C&UT) and SW integration testing (SWIT) into shorter iterations. Further, traditional developments produce volumes of documentation at each step down the waterfall. Documentation still needs to be created and maintained to satisfy the customer's SW maintenance requirements, record requirements for future procurements and competitions and to support corporate SEI requirements. Two additional stories need to be added to the initial iteration: one to determine the base set of standards for in-line documentation considering the emphasis on refactoring and a second to determine the necessary, minimum and sufficient set of maintenance documentation required to meet customer maintenance and corporate goals. As an example, in addition to the refinement of the story, a documented output of the iteration planning is a list of engineers who participated, when and where it occurred, decisions reached and action items to be resolved. This list of documentation is the core for the documentation requirements through OOD to SWIT. As a part of iteration planning, if additional documents (*e.g.* Interface Control Documents, *etc.*), require updates, the customer can write and prioritize stories to update and maintain these documents.

Test & Quality Assurance (QA)

As previously described, the test organization will be clearly effected. Adopting XP does not eliminate the need for separate test and QA organization. However, test and QA engineers will be integrated directly into the develop process as SWEs. The remaining test and QA organization, albeit with a reduced staff, will still conduct operational acceptance testing and evaluation (OAT&E), and end-to-end tests which require manual evaluation. Their function would be to test the stories and regression test the system (against large and varying known data sets). Further, in order to facilitate the automation of OA&TE tests and regression tests, the test and QA organizations should be financially incentivized. Automated tests will be constructive to the development, test and direct support organizations.

Fiefdoms

The first rule of any bureaucracy is that it is self-sustaining. A BDUF system comes with a big staff and over time it's own culture. Part of that culture is the creation of fiefdoms within the program organization. Adopting XP will radically change the functions of the organization within the program and consequently change the staff and funding profiles of the organizations. Bureaucracy (by way of the current culture and middle management) will want to protect itself from change. Buy-in by program management, and adoption by management (both customer and program) is an early important step to foster change in the culture and bureaucracy.

6 CONCLUSION

The transition to XP for O&M development is financially, technically and politically feasible. But the transition, as well as the expectations of the staff, customer and users need to by mapped and managed: especially for a customer who is averse to risk. Early defect detection and repair is cost effective and lowers the risk of unplanned perturbation to the operational baseline. XP gives us the methodology.

It is critically important while implementing the roadmap to remember: the system is the product and the people make the system.

REFERENCES

1. Pressman, Roger S. Software Engineering, A Beginners Guide. McGraw Hill, 1988, 13.
2. A detailed presentation of the EVMS can be found on-line at: http://www.acq.osd.mil/pm/.
3. A detailed presentation of SEI can be found on-line at: http://www.sei.cmu.edu/.
4. Berkowitz, Bruce D. and Goodman, Allen E. Strategic Intelligence for American National Security. Princeton University Press, 1991 (reprint), 54.
5. A detailed presentation of CMMI can be found on-line at: http://www.sei.cmu.edu/cmmi/.