

Running Head: ORGANIZATIONAL CHANGE PLAN FOR REENGINEERING THE
SOFTWARE DEVELOPMENT PROCESS

Organizational Change Plan for Reengineering the Software Development Process

Joseph C Thomas

Regent University School of Leadership Studies
LEAD612 Organizational Modification Strategies

February 23, 2004

I. Executive Summary

The objective of this change plan is to gain approval to transform how the company builds application software. Approval of this change plan launches a project to make substantial changes in the Information Technology Services organization by instituting a continuous improvement process that achieves real results. The business opportunity is to improve satisfaction by delivering the right product to the business customer and realize cost savings by delivering application systems on time with minimal defects. These primarily qualitative benefits are difficult to assign a dollar value, but with an annual budget of \$40M to \$50M for application software development and maintenance as well as the company strategy of custom building a substantial portion of the application portfolio, the benefits are nonetheless significant.

Past efforts to introduce rigorous continuous improvement into software development processes have had limited success. This effort is different because it leverages a proven approach to effecting large scale change in organizations and capitalizes on lessons learned from previous change efforts. These factors minimize the associated risks while providing the basis to effectively leverage the company's investments in people, process, and technology to a greater degree than would be possible in other investment alternatives (such as making no changes, package implementation, outsourcing, or other change in strategy). Effective return on investment is easily monitored by clear visibility into progress of the effort with early opportunities to adjust priorities, change approaches, and even stop the effort if warranted.

The probability of success for this effort is maximized because of strong leadership (it is lead by a team consisting of several key individuals recognized in the company's leadership succession plan). All indications are that this effort is positioned to be the most successful continuous improvement change initiative in the history of the information technology area.

II. Introduction

Fowler and Rifkin (1990) of the Carnegie Mellon University Software Engineering Institute (SEI) state that “improving the process of software systems development and maintenance is the most reliable way to improve product quality” (p. 3). Executive leadership within both Information Technology Services (ITS) and the company recognize the need to raise the performance level of the Software Engineering organization by improving the quality, effectiveness, and efficiency of the process that delivers software solutions for the business. If performance is improved, then the company expects to improve customer satisfaction and realize substantial savings by delivering the right product on time with minimal defects.

Despite the best efforts of many well intentioned and hard working people, past attempts at organizational change in the Software Engineering organization have consumed large amounts of money and time with little appreciable results. This situation has to change or the company will be forced to search for more cost effective alternatives to in-house application software development.

Why has past efforts at organizational change in the Software Engineering organization failed to yield significant results? Large scale organizational change is difficult, but some companies are successful. Kotter & Cohen (2002) assert that “We fail at change efforts not because we are stupid, over controlled and unemotional beings, although it can seem that way at times. We fail because we haven’t sufficiently experienced highly successful change. Without that experience, we are too often left pessimistic, fearful, or without enough faith to act.” (p. 13).

This is a plan for launching an organizational change effort to transform the current software development processes in use by the Software Engineering organization. What’s different about this plan is that it is based upon a model of successful change at other companies and it capitalizes on lessons learned to change the overall change strategy.

This change plan is presented as several components:

History of the Organization. This section presents background information about the company, ITS, and the Software Engineering organization.

Recommended Changes. This section presents the general approach within which specific change initiatives are pursued.

Analysis of the Change Process. This section presents a description of the overall change process within which each specific change initiative operates.

Specific Recommendations. This section presents specific change initiatives that implement the recommended changes.

Addressing Resistance to Change. This section presents reasons for resistance to change and describes application of several themes that can help in overcoming resistance to change.

Summary. This section presents a summary of the overall plan.

III. History of the Organization

The Software Engineering organization has reached a critical point in its history as internal and external pressures are combining to force significant change in how it operates if it is to survive and prosper. Company pressures such as market competition, regulatory changes, and the prospect of buy out or merger mean that the Software Engineering organization and ITS as a whole must be effective since the potential exists that the function could be outsourced. The company has recognized the need for software process improvements for some time. Previous efforts at improving organizational effectiveness of the software development process have cost a great deal of money and time and have been less than totally successful. The need for improvements continues to mount as the Software Engineering organization struggles with successfully implementing meaningful change in its software development processes.

The company is a one hundred year old company that is a leading provider of energy services across the United States. The company has about ten thousand employees while the ITS organization consists of about 700 people with a Software Engineering staff of about 450.

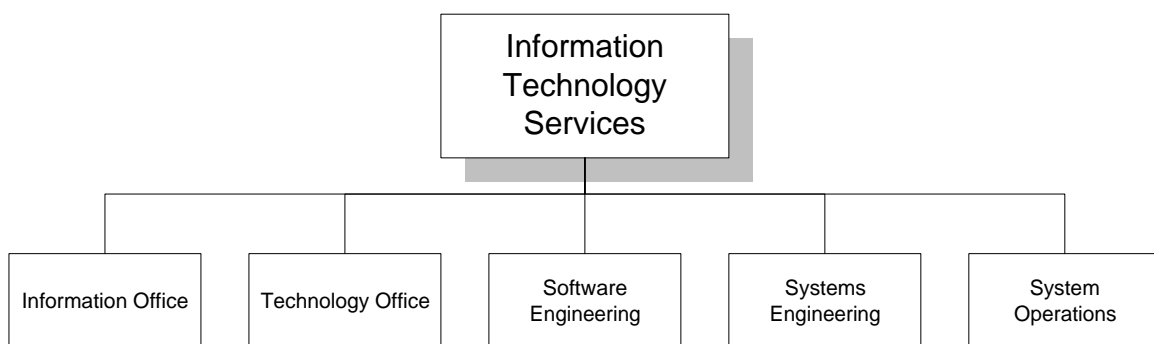


Exhibit 1 – The Information Technology Services Organization

As a component of Information Technology Services, the purpose of Software Engineering must be understood in the context of the purpose of ITS, its parent organization. ITS exists to quickly *deliver* high-quality solutions that provide direct business value to the company.

Software Engineering exists to efficiently *create* high-quality solutions that upon delivery provide direct business value to the company. Software Engineering is the focal point of the delivery process in that it creates the solution.

The purpose of Software Engineering must also be understood in the context of its peer organizations. Exhibit 1 depicts the relationships between organizations within ITS. The Information Office organization is the primary customer of Software Engineering. Although business people within the company are the ultimate end user of the solutions that Software Engineering creates, the Information Office is the primary customer because they are the interface to and represent the interests of the business organizations. The Technology Office provides the interface to the technology industry by researching, evaluating, and selecting current and emerging technology for appropriate application to business need. System Engineering designs and implements hardware, network, and system software solutions. System Operations operates the data center and production application systems. The Technology Office, System Engineering, and System Operations organizations team with Software Engineering in creating and delivering solutions that provide value to the business.

Although the software development organization is largely successful, improving quality and business value requires operating at an increased level of effectiveness. Two years ago, ITS management launched an initiative to become compliant with the Software Engineering Institute's (SEI) Capability Maturity Model (SW-CMM) (see www.sei.cmu.edu/cmm). This change effort was very expensive yet yielded few tangible results. This change plan represents a new effort to pursue process improvements, but it differs subtly but substantially from its predecessor because of the shift in strategy described in the balance of this document.

IV. Recommended Changes

As previously explained, the Software Engineering organization needs to raise the level of its performance to improve the quality, effectiveness, and efficiency of its software development process. Those best practices and lessons learned that exist must be codified so that the organization as a whole leverages its knowledge and experiences and can operate consistently and at a higher level of performance. Once existing best practices and lessons learned are codified, additional and new opportunities can be pursued. This section describes the general strategy to initiate the changes to begin the progression of the Software Engineering organization toward its performance improvement goals.

Software Engineering performance goals include improving Cost Performance Index (CPI) and Schedule Performance Index (SPI) of software projects, improving the ITS customer satisfaction index, reducing rework due to missed requirements, reduce defect rates, and reducing production system failure due to application design flaws.

The previous change effort focused on raising the performance level of the software development process as a whole to achieve the ideal state represented by a targeted level of the CMM. The CMM was used as a baseline comparison against current practice to perform a gap analysis. Then efforts were begun to address the gaps to achieve the level of practice represented by the CMM.

This new change plan is not prescriptive regarding the desired end state of software development processes; rather, it establishes the processes necessary for providing guidance, determining priorities, developing standards and guidelines, and establishes a continuous improvement process intended to go on indefinitely. The initial series of projects described in a subsequent section are initiated over a six month period. Each project is planned so that an initial version of its deliverables are completed and implemented within a short time period of three to

six weeks. Some projects may produce several versions of their artifacts during their initial period. Each project establishes a work group that forms the basis for an extended continuous improvement cycle that is repeated periodically (monthly, quarterly, twice a year, etc.). In total, the initial phase of the effort is planned to finish in an eight to twelve month period.

Exhibit 2 presents a conceptual overview of the timeline associated with the projects initiated by this change plan. Each of the eight projects (labeled X.0) represents an initial effort to produce one or more artifacts associated with a particular software development area. Subsequent projects (labeled X.1, X.2, X.3) represents the continuous improvement effort associated with the artifacts of a particular software development area. Although this change plan identifies specific change initiatives, the exact projects and their schedules are determined subsequent to launching the change effort.

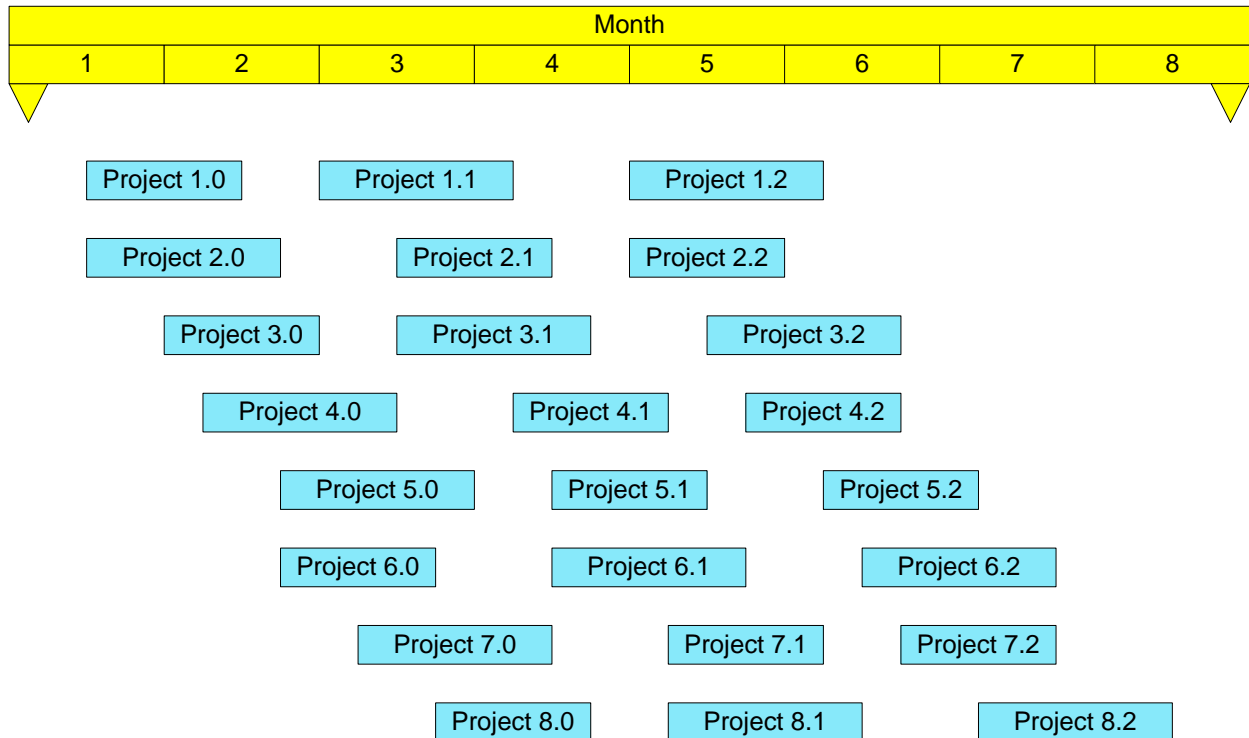


Exhibit 2 – Conceptual Overview of Timeline for Change Plan Initiatives

The strategy for implementing change has several key components:

- Guiding Leadership Team
- Practitioner Driven Change Makes It Real
- Little Talk, Lots of Action
- Project Manager Focal Point of Accountability
- Visualization of Success With Celebration
- Leverage What We've Got; Meet Them Where They're At
- Industry Standard Processes
- Self-Subscription Means Minimal Direct Budget
- Project and Artifact Oriented
- Process Selectivity and Prioritization
- Progressive Elaboration
- Principle Based

Each of these strategy components are described in the following paragraphs.

Guiding Leadership Team. The change effort is lead by this team. According to Kotter & Cohen (2002), a powerful guiding leadership team has two primary characteristics: “it is made of the right people, and it demonstrates teamwork” (p. 43). The guiding leadership team consists of key formal and thought leaders in the Software Engineering organization. The membership in this group is limited but flexible and inclusive as it reaches out to the Software Engineering group at large in an effort to identify and enlist individuals with expertise in particular disciplines and passion for improving the software development process. This team meets weekly to build consensus and teamwork as they initiate and oversee all projects associated with the change effort.

Practitioner Driven Change Makes It Real. The project work that implements specific change initiatives enlists individuals in the Software Engineering organization who have expertise and passion around particular process areas. Project teams leverage practitioner experiences and involve many people making individual contributions to build ownership of and buy in to results. The guiding leadership team is extremely sensitive to attempting to institute change that doesn't fit the reality of the people who do the work. Making the right changes to the right software engineering processes is carefully confirmed through getting the people who do the work to establish and review the improvements.

Little Talk, Lots of Action. Many required attendance meetings for the entire software engineering staff with lots of talk about the Software Capability Maturity Model and handing out doodads, posters, and reference cards characterized the previous software engineering change effort. There was a lot of hype, but in the end there was not much change affected.

This new change effort has a different communication plan that calls for no big announcements about the planned changes. Instead, changes are communicated at a grassroots level through the various project teams and review groups created to work on specific change initiatives. Initially, this process reaches a group of about 80 key people in the Software Engineering organization.

As various change improvements are completed, they are also communicated through the Software Engineering practice groups. Practice groups are the line organizations that are organized around specific software development disciplines: Project Management, Presentation Layer, Middle Layer, Persistence Layer, Technical Business Subject Matter Experts, Configuration Management, and Targeted Technologies. Each of the practice groups has meetings on a monthly or bimonthly basis. As each working group completes a particular

change, one of their tasks is to present the results at the appropriate practice group meetings to gain understanding and buy-in.

The Project Management practice provides an additional opportunity for communicating changes to the people in software development projects. Meetings with individual project managers provide the opportunity for one-on-one communication to leaders in direct day-to-day contact with most of the Software Engineering staff. Project managers are asked to make specific announcements to their teams regarding completed process improvement work that is relevant to the project teams' work. Additionally, offers are extended to the project manager to assist with preparation and presentation of these announcements.

As working groups complete results, an additional task is to get on the agenda and present to other regular organizational meetings such as the quarterly ITS wide meeting, weekly ITS leadership meeting, and staff meetings for the Technology Office, Information Office, Systems Engineering, and Systems Operations. The guiding group also watches for additional presentation opportunities and may create special targeted meetings if warranted.

These activities represent key leadership behaviors and communication processes as depicted in Exhibit 3. Each specific communication channel represents an opportunity for Software Engineering as well as ITS staff to receive change messages in different ways.

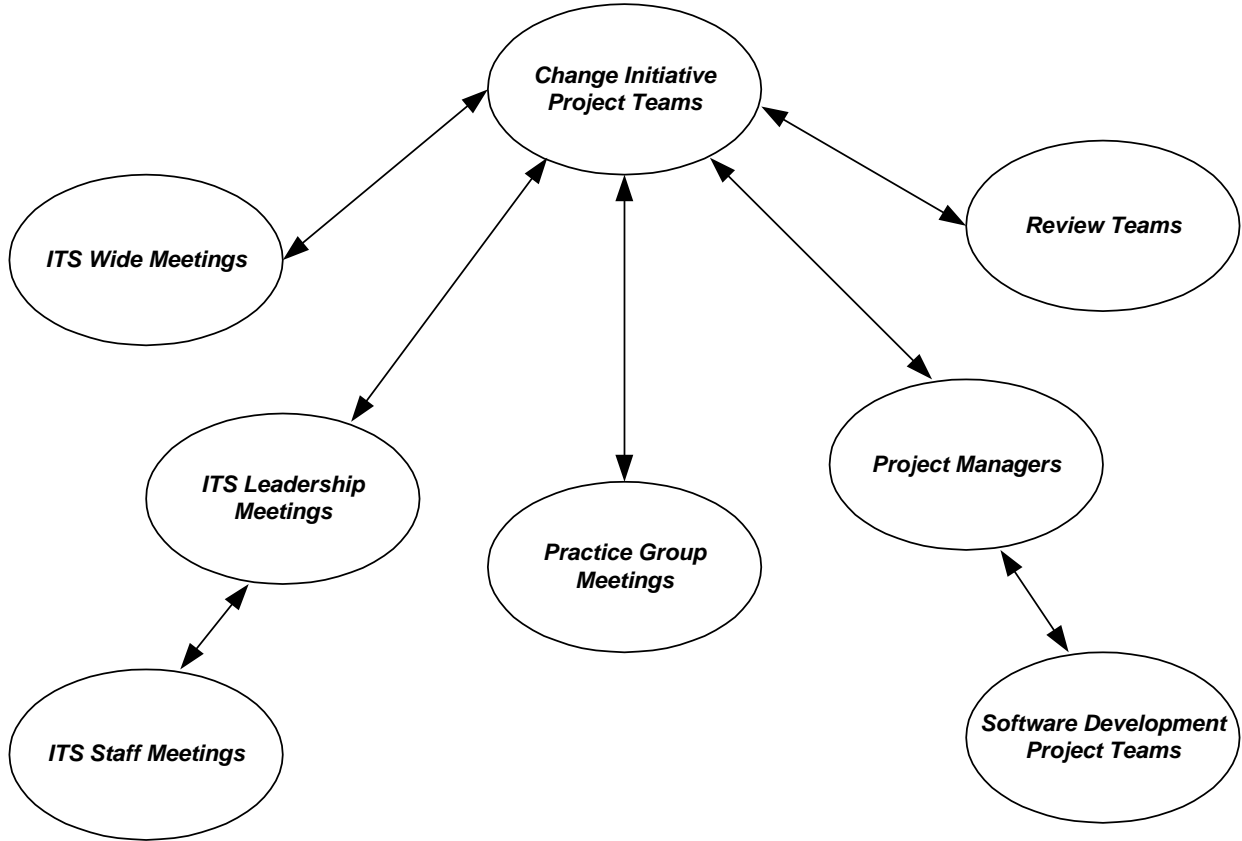


Exhibit 3 – Communication Plan Channels

Project Manager Focal Point of Accountability. Project managers provide direct leadership to software development project teams. Since project teams perform the work associated with software development processes, project managers provide the leadership that is needed to ensure that project teams understand and apply standards, guidelines, and other results of continuous improvement efforts. Project managers are accountable for project team adherence to standards and implementation of process improvements.

Visualization of Success With Celebration. Each individual initiative starts with a thoughtful description of what is different upon successful implementation. Understanding how success manifests itself and what to watch or monitor enables the team to know when success occurs. This provides for marking of milestones, celebrating victories, and sustaining enthusiasm over the long haul and in the face of obstacles. “In successful change efforts, empowered people

create short-term wins—victories that nourish faith in the change effort, emotionally reward the hard workers, keep the critics at bay, and build momentum” (Kotter & Cohen, 2002, p. 125).

This includes establishing and using formal measuring methods such as customer satisfaction surveys of business partners, surveys of Software Engineering staff, and quantitative monitoring and evaluation as appropriate.

Leverage What We’ve Got; Meet Them Where They’re At. Building upon processes that are in use and well understood by project teams and other organizational groups is much more likely to result in short term victories than trying to institute dramatic and all encompassing changes or by introducing new and unfamiliar processes. This is achieved through identification, evaluation, and modification of existing effective processes, procedures, and artifacts for incorporation into specific change initiative projects. Subsequent revisions of these initial standard processes enable the organization to incrementally improve rather than setting the bar at a level the organization is not prepared to achieve.

Industry Standard Processes. Leveraging industry standards reduces the expenditure of money, time, and energy as opposed to building customized or company specific resources that help improve software development processes. Several types of industry standards for software development process improvement were considered. See Appendix A for descriptions of the software engineering models considered.

The Software Engineering Institute’s Capability Maturity Model Integration (CMMI) (see www.sei.cmu.edu/cmmi/cmmi.html) was selected because it is an established and mature industry standard, does not represent a detailed software development methodology, is in the public domain (rather than a commercial product), and is well suited for software development process improvement. “The purpose of the CMMI Product Suite is to improve efficiency, return on investment, and effectiveness of process-improvement efforts by organizations using models

that integrate disciplines such as systems engineering and software engineering...” (*CMMI Frequently Asked Questions*, n.d.). CMMI coverage encompasses the entire spectrum of software development including system engineering, software engineering, integrated product and process development, and supplier sourcing (Chrissis, Konrad, & Shrum, 2003). See Appendix B for a description of the components of the CMMI model.

Inherent to the CMMI is a standard process improvement cycle adopted by this change effort (Fowler & Rifken, 1990):

1. Set expectations
2. Assess the current practice
3. Analyze the variance between expectations and practice
4. Propose changes
5. Plan the integration of the improvements
6. Implement the improvements
7. Perform the process as it is now defined
8. Start over

Self-Subscription Means Minimal Direct Budget. It is essential to enlist people to participate in change initiative projects that have a strong desire for improving the software engineering process. Self-subscription is striving for volunteer-based continuous improvement effort to which people choose to belong. The budget for the change effort consists primarily of the time of leadership team members; individual contributions from other people involved in special interest work groups and other teams are funded by the software development projects to which they belong.

Such contributions are recognized as above and beyond normal job duties and provide the basis for outstanding performance and associated rewards in addition to providing personal,

professional, and career development. Additionally, they provide the basis for the guiding leadership team to identify new leadership talent for both the change effort and ITS at large.

Project and Artifact Oriented. The work involved in the change effort consists of an ongoing series of individual projects initiated at different times to address specific goals according to priorities that the guiding leadership team establishes. Each project effort consists of a leader, a team, specific goals, and a definitive time frame. Multiple efforts operating simultaneously result in a stream of victories achieved over time. Project results are specific artifacts used by people in the Software Engineering organization. These artifacts are standards and guideline documents, checklists, process descriptions, step-by-step procedures, and other useful items.

Process Selectivity and Prioritization. Selection of specific software development processes for continuous improvement efforts is critical to ensure successful change. The change effort addresses specific high priority processes or process areas that are most likely to provide real value rather than trying to address the entirety of the software development process at one time. This approach limits the breadth of opportunity providing the change effort with a relatively narrow focus to maximize the likelihood of meaningful results while minimizing the risks of confusion and dispersion of effort to which Kotter & Cohen (2002) refers: “Because of the vary nature of large-scale change, much must be done to achieve the vision....When people feel urgency and are empowered to act, they can easily charge ahead on all fronts. With scattered attention, you might find the first unambiguous win [takes too long]” (p. 127).

The change effort selectively focuses first on areas that have pressing need, pockets of maturity that can be moved out into the organization at large, established processes, available subject matter experts possessing a high degree of passion around a particular area, or where quick and certain victories can be achieved.

Progressive Elaboration. Progressive elaboration is an iterative and incremental approach where deliverable characteristics are defined broadly at first then made more explicit and detailed as the effort continues (*A Guide to the PMBOK*, 2000). This approach delivers intermediate but meaningful results in short periods of time. Each effort builds upon its predecessor as well as provides for frequent priority adjustments so that changes yielding the most relevant benefits get addressed in a timely manner. This approach helps to realize focus similar to Process Selectivity and Prioritization, but limits the depth of opportunity instead of the breadth.

Principle Based. Rather than create volumes of detailed documentation describing legalistic rules and regulations, the emphasis is development of principles and standards that people apply intelligently to their particular situations. As Dee Hock, CEO emeritus of Visa International is widely quoted: “Simple, clear purpose and principles give rise to complex and intelligent behavior. Complex rules and regulations give rise to simple and stupid behavior” (*Dee Hock*, n.d.).

V. Analysis of the Change Process

Individual change initiatives launched as a result of implementing this change plan take the form of individual projects that are initiated by the guiding leadership team. The change process as a whole and each of these individual initiatives are similar in that they all go through a standard life cycle. This section describes that overall life cycle, describes key considerations for the change process, and identifies necessary leadership behavior. It is critical that all members of the guiding leadership team understand this approach.

Kotter's & Cohen's (2002) eight steps to affecting change in an organization represent the model life cycle of the change plan as a whole and each change initiative project. Kotter & Cohen (2002) assert "people change what they do less because they are given *analysis* that shifts their *thinking* then because they are *shown* a truth that influences their *feelings*" (p. 1). This is the single most important theme in Kotter's & Cohen's approach and is the overriding principle that guides all of the projects that initiate change.

Note that these eight steps represent a pattern that applies generally to the change process. The steps can occur repeatedly for each initiative, some could be skipped and returned to, or occur out of sequence because of multiple cycles or the particular needs for a given situation.

Kotter's & Cohen's (2002) eight steps to the change process are:

Step	Action	Behavior
1	Increase urgency	Raise the level of urgency and create a realization of the need for action so that the right people are enthusiastic about joining the effort. This creates what Tom Peters calls the "bias for action" (<i>Thomas J. Peters, n.d.</i>) necessary to make change happen. This occurs at all levels of leadership in the change effort.

Step	Action	Behavior
2	Build the guiding team	Form a group that is composed of key leaders (both formal leaders and thought leaders) and has an effective method of operating to guide the change process. Establish a guiding team with the right combination of formal authority, credibility and respect in the organization, skills, and connections.
3	Get the vision right	Establish the right vision and strategies to guide action for all change initiatives. This needs to be real and more than just budgets and timelines.
4	Communicate for buy-in	Communicate the visions and strategies for change with simple and heartfelt messages so people understand and have a gut-level feeling that vision is appropriate and achievable. Use repetition to ensure that the right people understand the right message is key.
5	Empower action	Mitigate obstacles that are preventing action focusing especially on neutralizing negative line leaders, dispelling misinformation, shoring up inadequate information systems, and promoting self confidence on the part of team members.
6	Create short-term wins	Ensure that meaningful victories are achieved quickly to create energy within the group, win over pessimists and cynics, and build momentum. Careful selection of initial projects and a well-managed process is key.
7	Don't let up	Continue with waves of change initiatives until the vision becomes reality despite setbacks and obstacles along the way. Don't let up and don't try to do too much at once.
8	Make change stick	Ensure that the change becomes "the way we do things around here" thus integrating change into the culture. This requires consistent and successful action over a sufficient time period.

The behavior of leadership is critical to affecting the change in this plan. It is leadership that guides and enables individual teams to work their way through the eight steps. Kotter & Cohen (2002) identify the core behaviors observed in successful change efforts in large-scale organizations as see—feel—change as opposed to analyze—think—change.

In analyze—think—change, information is first gathered and analyzed. Reports and presentations are then prepared. A rational argument builds the case for change and efforts are made to sell the idea or convince people of the right way to go. This new thinking then results in changed behavior. Kotter & Cohen (2002) assert that analyze—think—change has at least three

weaknesses: 1) in many cases, you don't need analysis to find the big truths; 2) analytical tools have limitations when all the facts are not certain which is often the case in today's fast moving world; and 3) even good analysis rarely results in significant motivation on the part of groups of people.

In see—feel—change, people are shown, often through dramatic actions, the nature of the problem or solution. This visualization of the situation leads to an instinctive response that reduces objections and obstacles to change. People then begin to believe in the necessity of the change and the necessity of finding a way to make it happen. This feeling results in ideas that change behavior. “Motivation is not a thinking word; it's a feeling word” (Kotter & Cohen, 2002, p. 13).

VI. Specific Recommendations

This section outlines the specific steps to take that collectively begin implementing the continuous improvement process and thereby ultimately changing how the Software Engineering organization goes about the process of software development.

The specific steps are:

- Form the Software Engineering Process Group (SEPG)
- Form the Software Engineering Process Steering Committee
- Conduct assessments of the software engineering process
- Set software engineering process improvement priorities
- Establish mechanism to launch special interest work groups
- Launch software development project oversight special interest work group
- Launch special interest work group for establishing a code of conduct for ITS professionals
- Launch special interest work group for process asset library
- Launch special interest work group for software development standards and guidelines
- Launch special interest work group for project management standards and guidelines
- Launch special interest work group for technical business subject matter expert standards and guidelines
- Launch special interest work group for metrics and measurements
- Launch special interest work group for software development tools
- Launch special interest work group for continuous improvement of process assets

Each of these steps is described in the following paragraphs.

Form the Software Engineering Process Group (SEPG). As described by Fowler & Rifken (1990), the SEPG acts as the focal point for all continuous improvement efforts for software engineering processes. It is comprised of people within the Software Engineering organization that are widely respected by superiors, peers, and subordinates as experts in one or more particular disciplines of software engineering. The SEPG is a collaborative team whose activities include:

- Gaining support and interfacing with all levels of company management,
- Assessing the state of software engineering in the organization,
- Establishing priorities for continuous improvement,
- Launching special interest work groups to address specific continuous improvement needs
- Monitoring results to ensure goal achievement.

The leader of the SEPG must be a high level formal leader in the organization, a champion for changing the software engineering practices, and an experienced practitioner in software engineering that has earned the respect of his peers and subordinates in the organization. Other members of the SEPG include organizational subject matter experts in project management and the entire life cycle of software development projects.

The Director of Software Engineering has signed up to be the leader of the SEPG. This group meets weekly in collaborative work sessions to review priorities, review the work of special interest work groups, and deal with issues to ensure forward progress.

The SEPG acts as Kotter's & Cohen's guiding team for the entire change process. In this capacity, it is their responsibility to understand and buy in to Kotter's & Cohen's eight steps of change process as well as ensure that the SEPG and its special interest work groups are following these steps in the course of their activities.

Form the Software Engineering Process Steering Committee. As described by Fowler & Rifken (1990), the steering committee is the means by which management guides and approves the work of the SEPG. It consists of ITS executive and line managers who act as the:

- Oversight body to approve the work of the SEPG,
- Policy setting body to facilitate and legitimize the work of the SEPG
- Liaison to other corporate quality and planning organizations.

This group meets monthly or bimonthly to review the status of the SEPG's process improvement efforts.

Conduct assessments of the software engineering process. As described by Fowler & Rifken (1990), assessments identify areas for improvement and stimulate the process improvement cycle. While all agree that improvements are needed, the SEPG conducts both formal and informal assessments to objectively understand where the real opportunities are and to help establish implementation priorities. Formal assessments include questionnaires, interviews, and other instruments to identify both strengths and weaknesses of the software development process.

Preliminary assessments have confirmed impressions that there are pockets of excellence within the organization to leverage, and that there are software development efforts that deviate from industry norms and the practices associated with successful company projects. Rather than a one-time effort, assessments are an ongoing periodic activity providing information and perspective that helps the SEPG establish priorities

Set software engineering process improvement priorities. Using information and perspective from formal and informal assessments as well as the experience of the group's members, the SEPG prioritizes the opportunities for process improvement. An appropriate approach to determine priorities is to create a matrix that identifies specific and relevant

opportunities for process improvement and categorizes them according to such criteria as potential value, estimated cost, degree of risk, etc. This is an ongoing effort revisited as accomplishments, experience, and necessary adjustments dictate. Preliminary results from such a prioritization process are the basis for several of the special interest work groups identified in this plan.

Establish mechanism to launch special interest work groups. A special interest work group is a sub team of the SEPG that conducts detail work associated with an improvement cycle for a particular process or functional area. The SEPG forms special interest work groups to do the process improvement work once they have established priorities.

The SEPG must recruit leaders for these efforts. In many but not all cases, SEPG members serve as leaders for the special interest work groups that are in their area of expertise. The special interest work group may in turn form their own subordinate special interest work groups to help with detail work or serve as a review body. This provides opportunity to reach into the organization to gain buy-in to specific change and to recruit additional leaders and team members. The SEPG also works with the special interest work groups to establish objectives and strategy while ensuring that they understand and apply Kotter's & Cohen's eight steps of change.

Launch software development project oversight special interest work group. The project oversight special interest work group establishes the means to hold project managers accountable for the results of their projects adhering to standards and guidelines as well as implementing the results of continuous improvement efforts. To accomplish this, it is essential to understand the software development projects that are scheduled, in progress, and completed. Project managers report project results to the oversight work group upon completion of major milestones including project closure. These project results typically take the form of compliance affirmations. The

project oversight group is responsible for negotiating the formal approval that project results are compliant with relevant standards.

A major responsibility of this work group is to make apparent to the organization at large that the behavior of project teams is actually realizing the vision of the SEPG. This is accomplished by measuring and reporting results of individual projects to the SEPG, ITS management, and the ITS organization as a whole. These results represent victories for the SEPG and its working groups and are the basis for celebration.

Launch special interest work group for establishing a code of conduct for ITS professionals. Essential to the change strategy is the notion of personal responsibility on the part of Software Engineering staff. The code of conduct provides the basis for making explicit desired behavior, performance, and business relationships of ITS employees. Each ITS professional has an obligation to the company, ITS' business partners, the ITS organization, those they lead, their group/team, and themselves in their day-to-day work.

This effort establishes a project and a leader for improving the processes associated with the development of the ITS professional code of conduct. The project begins by establishing a project team, then develops and implements an initial version of the code of conduct statement, and concludes with establishing the continuous improvement process for this functional area.

Launch special interest work group for process asset library. As described by Fowler & Rifken (1990), a key role of SEPG is to establish a central collection of critical information about software engineering processes. This library houses the results of process improvement work completed by special interest work groups and enables the rest of the organization to access the information on demand. The objectives of this group include building a user interface for the process asset library consisting of a website on the company intranet. Design goals of this website include ease of access by the practitioner to the information they need most.

This effort establishes a project and a leader for improving the processes associated with the development the process asset library. The project begins by establishing a project team, then develops and implements an initial version of the process asset library, and concludes with establishing the continuous improvement process for this functional area.

Launch special interest work group for software development standards and guidelines.

Every software development organization needs a set of standards and guidelines for designing and constructing software applications. Software Engineering strives to deliver quality software applications in a repeatable, consistent manner to help our business partners achieve success.

This effort establishes a project and a leader for improving the processes associated with software design and construction. The project begins by establishing a project team, then develops and implements an initial version of guidelines for the software design and programming process, and concludes with establishing the continuous improvement process for this functional area.

Launch special interest work group for project management standards and guidelines.

Project management is defining the work requirements (including quantity and quality), establishing resource needs, tracking progress, comparing actual outcomes to predicted results, and making adjustments as the project work unfolds. (Kerzner, 1998, p. 3). Project management provides the leadership at the level in the organization where the work is done and is therefore a critical success factor for software development projects. The Software Engineering organization has enjoyed many successful projects resulting in large part from the leadership provided by many good project managers, but this success has usually been the result of individual effort rather than an established organizational culture. This effort seeks to establish best practices for project management so all project managers can learn from the experiences of each other thereby

ensuring that as an organization we leverage the lessons that have been learned by individual project managers.

This effort establishes a project and a leader for improving the processes associated with software project management. The project begins by establishing a project team, then develops and implements an initial version of guidelines for the software project management process, and concludes with establishing the continuous improvement process for this functional area.

Launch special interest work group for technical business subject matter expert standards and guidelines. The Software Engineering organization several years ago combined the roles of business systems analyst and software tester to form the single role of technical business subject matter expert. The organization recognized that proper testing of a complex, business application requires extensive knowledge of the business and that software testers often had as much or more knowledge about the business and its operations than business systems analysts or our business partners.

This effort establishes a project and a leader for improving the processes associated with technical business subject matter experts. The project begins by establishing a project team, then develops and implements an initial version of guidelines for the technical business subject matter expert process, and concludes with establishing the continuous improvement process for this functional area.

Launch special interest work group for metrics and measurements. Metrics and measurement involves specifying measurement objectives to meet information needs, specifying and implementing approaches for data capture, storage, analysis, and reporting, and providing end results for use in decision making (Chrissis, Konrad, & Shrum, 2003).

This effort establishes a project and a leader for improving the processes associated with metrics and measurement. The project begins by establishing a project team, then develops and

implements an initial version of the metrics and measurement process, and concludes with establishing the continuous improvement process for this functional area.

Launch special interest work group for software development tools. There are many different software tools in use across the Software Engineering organization. It is the desire of the organization to provide the right tools that people need to be effective, but also to minimize and control software license costs while effectively managing acquisition and distribution of software licenses.

This effort establishes a project and a leader for improving the processes associated with software development tool management. The project begins by establishing a project team, then develops and implements an initial version of the software development tool management process, and concludes with establishing the continuous improvement process for this functional area.

Launch special interest work group for continuous improvement of process assets. Each of the special interest work groups launched by these individual initiatives provides focus on improving the process for a particular high priority process area and results in production of one or more artifacts. A major risk for these artifacts is that they become “shelfware;” that is, the investment in them atrophies over time because the practitioners in the Software Engineering organization find them irrelevant and therefore they do not become integrated into day-to-day work activities. This can also happen when the artifacts are not regularly improved to keep pace with changes in technology, processes, and organizational structure.

The key to this approach is personal responsibility by a named individual. For each special interest work group to establish a continuous improvement process for their functional area, they must enlist a person that possesses subject matter expertise for a specific period of time to lead the continuous improvement effort.

Specific responsibilities of the leader of a continuous improvement process for a functional area are:

1. Negotiate an agreement with the SEPG that establishes artifact objectives, improvement cycles, and term of service.
2. Enlist qualified individuals to participate in the special interest work group for the artifact.
3. Manage the artifact feedback loop by monitoring the artifact mailbox, classifying emails, talking with contributors, and selecting recommended opportunities for improvement.
4. Schedule, plan, and set agenda for special interest work group collaboration sessions.
5. Drive the artifact review and production cycle.
6. Form and solicit input about artifact improvements from a review group.
7. Gain approval from oversight bodies (SEPG, practice leaders, and other special interest work groups).
8. Make presentations and otherwise promote artifact revisions consistent with the change communication plan.
9. Identify and grow a replacement leader from the artifact special interest work group.
10. Enlist a replacement leader and help establish objectives for next term of service.

This effort establishes a project and a leader for improving the continuous improvement process. The project begins by establishing a project team, then develops and implements an initial version of the continuous improvement process, and concludes with establishing the continuous improvement process for this functional area.

VII. Addressing Resistance to Change

Strong resistance to change is the norm rather than the exception. This section briefly outlines reasons for such strong resistance and describes application of several themes that can help in overcoming resistance to change.

Kotter & Cohen (2002) assert that the “biggest challenge in the [organizational change] process is changing people’s behavior” (p. 170). There is a myriad of reasons why people resist change: fear of the unknown, comfort with the status quo, cynicism about the need to change, doubts about the ability to change, conflicting goals, confusion about objectives, conflict with individual values, and conflict with the way people think things should be done (McNamara, 1999). O’Toole (1996) examines the literature on the reasons for resistance to change and identifies thirty three different popular hypotheses, but boils it down to two primary reasons that are particularly applicable to the ITS organization: organizational culture and oligarchic control.

An easily understandable definition of organizational culture is “the way we do things around here.” Successful organizations typically have a set of operating assumptions that dictate how people behave—the behaviors that have made the organization successful are the behaviors the people tend to repeat (O’Toole, 1996). Such a culture has a powerful inertia that is difficult to overcome.

Oligarchic control is when a relatively small group manages the entire organization (Merriam-Webster, n.d.). People tend to follow the dictates of this leadership group because they establish efficiency and order as the ultimate goal; over time, this typically results in a bureaucracy optimized to perform a standard and predictable set of activities. This can be particularly applicable to a large and historied organization. Such an organization restricts the freedom of individuals and inhibits innovation (O’Toole, 1996).

The Software Engineering Process Group (SEPG) as the team guiding the change process, with the assistance of the Software Engineering Process Steering Committee, monitor results of the change effort and its various change initiatives to determine whether there is sufficient progress. The concepts of organizational culture and oligarchic control are useful to provide insight for determining adjustments to the activities of the change effort when warranted.

Head (1997) identifies several themes to ensure the success of change efforts: ensuring that the people closest to the work are involved in the change effort, ensuring that the change initiatives are relevant as well as properly aligned with company and ITS strategic objectives, and establishing continuous learning and improvement systems to sustain meaningful change. The first two themes address changing the organizational culture (how ITS does software engineering) by engaging the right people on the right work processes. The third theme addresses the distribution of leadership and decision making to the people who do the work rather than those in formal organizational leadership positions.

Head's three themes are inherent to the overall change plan, but can be difficult to maintain over the extended duration of the change effort. Lack of sufficient progress indicates a need to evaluate how well the SEPG and the special interest work groups are incorporating these themes in their work and may require creative application or redoubling efforts in one or more of these theme areas.

VIII. Summary

As the French author Alexandre Dumas (1802-1870) is oft quoted “Nothing succeeds like success” (*Quotations*, n.d.). As the Software Engineering organization successfully changes itself through a series of small but meaningful changes, subsequent change can be sustained as one success builds upon another.

The main theme of this change plan is demonstrated incremental progress through application of a change process proven to work at many large organizations. This is accomplished by an organizationally bottom up effort led by a small team of the company’s proven leaders and subject matter experts. Change is effected by appealing to the gut instinct of people throughout the Software Engineering organization rather than relying on the analysis of facts and figures. Resistance to change is dealt with by monitoring progress to identify when resistance occurs and ensuring that the right people are engaged on the right work processes and that they are empowered to make decisions.

Effective implementation of a continuous improvement process raises the performance level of the Software Engineering organization by improving the quality, effectiveness, and efficiency of the process that delivers software solutions to the business. Raising the level of performance enables delivery of the right product on time with minimal defects which leads to customer satisfaction improvements and cost reductions. This is a significant opportunity with little accompanying investment risk as the approach to change requires a small budget and progress is readily determinable so as to facilitate changing priorities or stopping altogether. Given these considerations, the recommendation is to approve this change plan and launch the project to institute a continuous improvement process for Software Engineering.

Appendix A: Software Engineering Models

This appendix summarizes several software engineering models and methods considered for application to the change effort.

Adaptable Process Model (R. S. Pressman & Associates, 2001). This is a commercial product designed to be customized by a software engineering organization for their particular use. It is methodology oriented tool that provides wizard-like configuration of a set of activities for projects of particular types.

Electronic Industries Alliance (EIA) Systems Engineering Capability Model (SECM) (EIA/IS-731). This is both a model and an assessment method that addresses the improvement of systems engineering capability. “Systems engineering is an inter-disciplinary approach and means to enable the realization of successful systems. In this context, systems engineering is not limited to what either systems engineering organizations or systems engineers do. Rather it is the interaction of many people, processes, and organizations resulting in the accomplishment of the required activities” (EIA/IS-731, 1999, ¶ 5). This is an interim standard, and since its development has been incorporated into the CMMI.

Enterprise Process Improvement Collaboration (EPIC) Integrated Product Development Capability Maturity Model (IPD-CMM). This is a model that describes the essence of integrated product development and provides an approach for process improvement (*Integrated Product*, 1997). This model has been incorporated into the CMMI (Chrissis, Konrad, & Shrum, 2003).

IBM Rational Unified Process (RUP) (IBM, n.d.). This is “a configurable software development process platform that delivers proven best practices and a configurable architecture” that “enables you to select and deploy only the process components you need for each stage of your project” (IBM, n.d., ¶ 1 & 2). This is a commercial product that describes a detailed software development methodology.

Information Technology - Software Process Assessment ISO/IEC TR 15504 (ISO/IEC TR 15504, n.d.). This is similar to the CMMI, but is an international capability assessment framework that is an emerging standard.

Software Engineering Body of Knowledge (SWEBOK) (Bourque, P., Dupuis, R., & Tripp, L., 2001). This is primarily a reference work that summarizes software engineering practices and provides references to them in the industry literature.

Software Engineering Institute (SEI) Capability Maturity Model for Software (CMM or SW-CMM) (Capability Maturity Model, n.d.). The CMM has been the de facto standard for assessing and improving the processes of software development for many years. The Software Engineering Institute was formed in conjunction with the United States federal government for this purpose. The CMM has been superseded by the CMMI.

Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI) (CMMI General Information, n.d.). This is the most popular representation of process improvement models in the industry. SEI advocates them as “the best process improvement models available for product and service development and maintenance. These models build on and extend the best practices of the Capability Maturity Model for Software (SW-CMM), the Systems Engineering Capability Model (SECM), and the Integrated Product Development Capability Maturity Model (IPD-CMM)” (CMMI General Information, n.d, ¶ 2).

Appendix B: CMMI Process Areas

This appendix summarizes the CMMI process areas, goals, and generic practices (Chrissis, Konrad, & Shrum, 2003).

<i>Process Area</i>	<i>Description</i>
Causal Analysis and Resolution	Identify causes of defects and other problems and take action to prevent them from occurring in the future.
Configuration Management	Establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.
Decision Analysis and Resolution	Analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria.
Integrated Project Management	Establish and maintain the project and the involvement of the relevant stakeholders according to an integrated and defined process that is tailored from the organization's set of standard processes.
Integrated Supplier Management	Proactively identify sources of products that may be used to satisfy the project's requirements and to manage selected suppliers while maintaining a cooperative project-supplier relationship.
Integrated Teaming	Form and sustain an integrated team for the development of work products.
Measurement and Analysis	Develop and sustain a measurement capability that is used to support management information needs.
Organizational Environment for Integration	Provide an Integrated Product and Process Development (IPPD) infrastructure and manage people for integration.
Organizational Innovation and Deployment	Select and deploy incremental and innovative improvements that measurably improve the organization's processes and technologies. The improvements support the organization's quality and process-performance objectives as derived from the organization's business objectives.
Organizational Process Definition	Establish and maintain a usable set of organizational process assets.
Organizational Process Focus	Plan and implement organizational process improvements based on a thorough understanding of the current strengths and weaknesses of the organization's processes and process assets.
Organizational Process Performance	Establish and maintain a quantitative understanding of the performance of the organization's set of standard processes in support of quality and process-performance objectives, and to provide the process performance data, baselines, and models to quantitatively manage the organization's projects.
Organizational Training	Develop the skills and knowledge of people so they can perform their roles effectively and efficiently.
Product Integration	Assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product.
Project Monitoring and Control	Provide an understanding of the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.
Project Planning	Establish and maintain plans that define project activities.
Process and Product Quality Assurance	Provide staff and management with objective insight into processes and associated work products.

<i>Process Area</i>	<i>Description</i>
Quantitative Project Management	Quantitatively manage the project's defined process to achieve the project's established quality and process-performance objectives.
Requirements Development	Produce and analyze customer, product, and product-component requirements.
Requirements Management	Manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products.
Risk Management	Identify potential problems before they occur so that risk-handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives.
Supplier Agreement Management	Manage the acquisition of products from suppliers for which there exists a formal agreement.
Technical Solution	Design, develop, and implement solutions to requirements. Solutions, designs, and implementation encompass products, product components, and product-related life-cycle processes either singly or in combination as appropriate.
Validation	Demonstrate that a product or product component fulfills its intended use when placed in its intended environment. ("You built the right thing.")
Verification	Ensure that selected work products meet their specified requirements. ("You built it right.")

<i>Generic Goal and Generic Practices</i>	<i>Description</i>
Achieve Specific Goals	The process supports and enables achievement of the specific goals of the process area by transforming identifiable input work products to produce identifiable output work products.
Perform Base Practices	Perform the base practices of the process to develop work products and provide services to achieve the specific goals of the process area.
Institutionalize a Managed Process	The process is institutionalized as a managed process.
Establish an Organizational Policy	Establish and maintain an organizational policy for planning and performing the process.
Plan the Process	Establish and maintain the plan for performing the process.
Provide Resources	Provide adequate resources for performing the process, developing the work products, and providing the services of the process.
Assign Responsibility	Assign responsibility and authority for performing the process, developing the work products, and providing the services of the process.
Train People	Train the people performing or supporting the process as needed.
Manage Configurations	Place designated work products of the process under appropriate levels of configuration management.
Identify and Involve Relevant Stakeholders	Identify and involve the relevant stakeholders of the process as planned.
Monitor and Control the Process	Monitor and control the process against the plan for performing the process and take appropriate corrective action.
Objectively Evaluate Adherence	Objectively evaluate adherence of the process against its process description, standards, and procedures, and address noncompliance.
Review Status with Higher Level Management	Review the activities, status, and results of the process with higher level management and resolve issues.
Institutionalize a Defined Process	The process is institutionalized as a defined process.
Establish a Defined Process	Establish and maintain the description of a defined process.

<i>Generic Goal and Generic Practices</i>	<i>Description</i>
Collect Improvement Information	Collect work products, measures, measurement results, and improvement information derived from planning and performing the process to support the future use and improvement of the organization's processes and process assets.
Institutionalize a Quantitatively Managed Process	The process is institutionalized as a quantitatively managed process.
Establish Quantitative Objectives for the Process	Establish and maintain quantitative objectives for the process that address quality and process performance based on customer needs and business objectives.
Stabilize Subprocess Performance	Stabilize the performance of one or more subprocesses to determine the ability of the process to achieve the established quantitative quality and process-performance objectives.
Institutionalize an Optimizing Process	The process is institutionalized as an optimizing process.
Ensure Continuous Process Improvement	Ensure continuous improvement of the process in fulfilling the relevant business objectives of the organization.
Correct Root Causes of Problems	Identify and correct the root causes of defects and other problems in the process.

References

- Bourque, P., Dupuis, R., & Tripp, L. (Ed.). (2001). *Guide to the software engineering body of knowledge*. Los Alamitos, CA: IEEE Computer Society. Retrieved February 16, 2004 from <http://www.swebok.org/>
- Capability Maturity Model for Software (SW-CMM)*. (n.d.). Retrieved February 21, 2004 from <http://www.sei.cmu.edu/cmm/>
- CMMI Frequently Asked Questions*. (n.d.). Retrieved January 27, 2004 from <http://www.sei.cmu.edu/cmmi/adoption/cmmi-faq.html>
- CMMI General Information*. (n.d.). Retrieved February 16, 2004 from <http://www.sei.cmu.edu/cmmi/cmmi.html>
- Chrissis, M., Konrad, M., & Shrum, S. (2003). *CMMI: Guidelines for Process Integration and Product Improvement*. New York: Addison-Wesley.
- Dee Hock*. (n.d.). Retrieved February 21, 2004 from <http://dqs.worldatwar.org/robots/1514.html>
- EIA/IS-731 Systems Engineering Capability Model*. (1999). Retrieved February 19, 2004 from <http://www.incose.org/stc/news731.htm>
- Fowler, P. & Rifken, S. (1990). *Software Engineering Process Group Guide*. Pittsburgh: Software Engineering Institute Carnegie Mellon University
- Head, C. (1997). *Beyond corporate transformation: a whole systems approach to creating and sustaining high performance*. Portland, OR: Productivity Press.
- IBM. (n.d.). Rational Unified Process. Retrieved February 16, 2004 from <http://www-306.ibm.com/software/awdtools/rup/>
- Integrated Product Development Capability Maturity Model Overview Briefing*. (1997). Retrieved February 19, 2004 from http://www.secat.com/download/locked_pdf/IPDovrvw_lkd.pdf
- ISO/IEC 15504 - An Emerging Standard on Software Process Assessment*. (n.d.). Retrieved February 16, 2004 from <http://www.sei.cmu.edu/iso-15504/moreinformation/seimotives.html>
- ISO/IEC TR 15504*. (n.d.). Retrieved February 16, 2004 from <http://isospice.com/standard/tr15504.htm>
- Kerzner, H. (1998). *Project Management – A Systems Approach to Planning, Scheduling, and Controlling*. (6th ed.). New York: Wiley & Sons.
- Kotter, J. & Cohen, S. (2002). *The Heart of Change: Real Life Stories of How People Change Their Organizations*. Boston, MA: Harvard Business School Press.

McNamara, C. (1999). *Basic Context for Organizational Change*. Retrieved March 30, 2004 from <http://www.mapnp.org/library/mgmt/orgchnge.htm>

Merriam-Webster. (n.d.). Retrieved March 30, 2004 from <http://www.m-w.com/>

O'Toole, J. 1996. *Leading change: the argument for values-based leadership*. New York: Ballantine Books.

A Guide to the Project Management Body of Knowledge (PMBOK Guide). (2000 ed.). Newtown Square, PA: Project Management Institute.

Quotations and Famous Quotes. (n.d.). Retrieved February 21, 2004 from <http://www.quoteworld.org/>

R.S. Pressman & Associates. (2001). *Adaptable Process Model*. Retrieved February 16, 2004 from <http://www.rspa.com/index.html>

Thomas J. Peters. (n.d.). Retrieved February 21, 2004 from http://www.cyber-nation.com/victory/quotations/authors/quotes_peters_thomasj.html

Welcome to the CMMI Web Site. (n.d.). Retrieved January 27, 2004 from <http://www.sei.cmu.edu/cmmi/cmmi.html>