

Organizational Structure for Innovative Software Development

Joseph C. Thomas

Regent University Center for Leadership Studies

LEAD606 Strategic Vision and Organizational Effectiveness

30-Sep-2002

Introduction

The computer software development industry has been struggling to effectively operate processes that reliably and predictably complete projects on time and within budget. Oft quoted research statistics from the Standish Group (1995) are staggering: 31% of all projects are cancelled before they are completed. 53% of projects overrun their budgets by 189% of their original budget. Only 16% of projects complete on time and within budget, and in large organizations, this figure is only 9%. To address some of the core problems, many software development organizations are beginning to successfully apply innovative new approaches known as lightweight or agile methods (Fowler, 2002) and extreme programming. These approaches are characterized by small teams, simple planning and design methods, test driven development, continuous design improvement, continuous integration, and sustainable pace (Jeffries, 2001).

This paper outlines the implementation of a new organization structure for the Software Engineering component of the Information Technology Services (ITS) organization at a Fortune 500 company. This new organization supports implementation of these innovative approaches and represents a significant change from an existing structure that is largely bureaucratic and not working well for these new approaches. This description of the Software Engineering organization focuses on the portion of the information technology services organization that builds application software systems. Included for the Software Engineering organization is a statement of purpose, a description of the organization structure, environmental contingencies, and necessary characteristics of the organization's leader.

Statement of Organization Purpose

The statement of organization purpose is expressed as two components: mission and vision. Mission is the “purpose [or] reason for being” and vision is the “picture or image of the future we seek to create” (Senge, 1998).

The ITS Software Engineering Mission Statement

The mission of Software Engineering is, in partnership with the Technology Office and the Information Office, to be the leading supplier to the company in creating high quality solutions, delivering them quickly, and providing direct business value. Although business people within the company are the ultimate end user of the solutions that the Software Engineering organization delivers, the Information Office is the primary customer because they are the interface to and represent the interests of the business.

The ITS Software Engineering Vision Statement

Software Engineering achieves its mission through project teams that exhibit key tenets of embracing change, encouraging collaboration, prioritizing value, and resolving risk early. Project teams are groups of people that come together to deliver a solution that has a real and measurable return on investment to the business. Embracing change is permitting customers to evolve their requirements as they incrementally gain insight into their real needs. Encouraging collaboration is continuous learning and self-organization within the project team and between all project stakeholders. Prioritizing value is delivering simple solutions that capitalize on only the essential aspects of delivering business value. Resolving risk early is front loading project work with the most difficult or uncertain aspects of the solution.

Vision

Project teams that:

- Embrace Change
- Encourage Collaboration
- Prioritize Value
- Resolve Risk Early

Benefits to Customers, Employees, and Society

The benefits to customers of pursuing the mission and achieving the vision include better control over investment dollars through reduced cost and lower risk, increased visibility into project results, rapid response to changing needs, innovative problem solving and solution creation, delivery of solutions that build on incremental accomplishments, and quick revelation of issues for resolution or even project cancellation. The benefits to employees include high morale and job satisfaction through improved attention to professional and career growth, enlistment into preferred job assignments, and successful project work. The benefits to society at large include more productive assets for increasing shareholder value and philanthropic contribution as well as happier families through a more satisfied and productive workforce.

Organization Structure

Figure 1 depicts the Software Engineering organization in context with the balance of the ITS functional organization. Table 3 summarizes the purpose of each component.

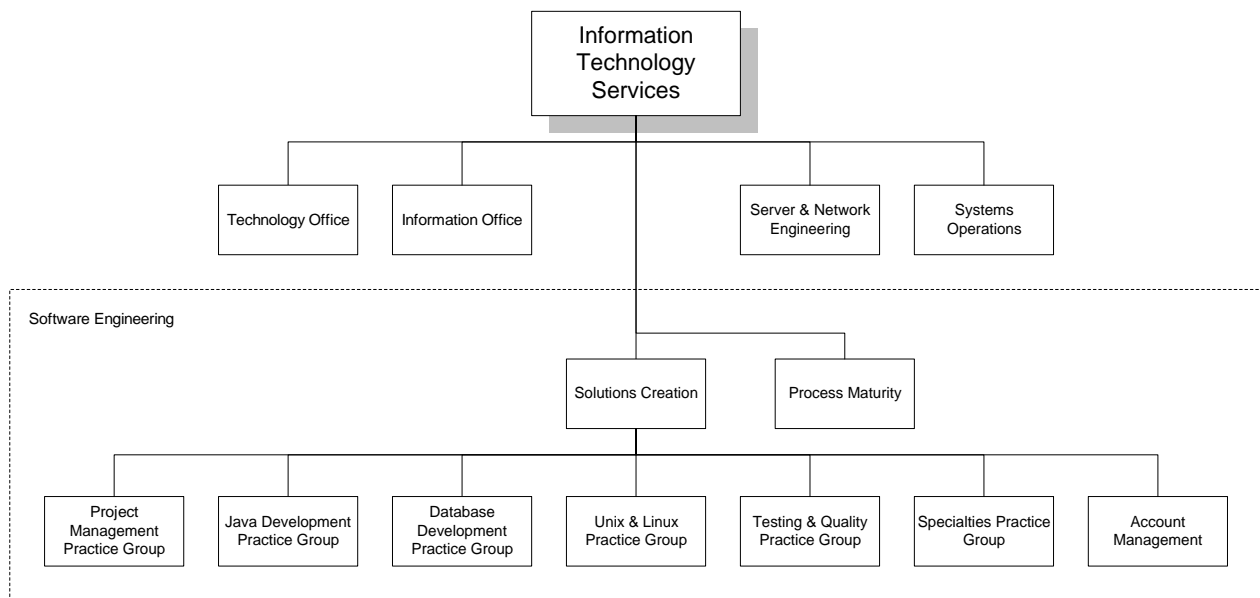


Figure 1: Functional Organization

A key strategy for Software Engineering is the use of a matrix organization for project work, as Figure 2 depicts. This form is ideally suited for an organization that is primarily focused on project work (Kerzner, 1998; p. 110). Of the many advantages of this organizational form (Kerzner, 1998; p. 115), the most important in this situation center around the flexibility of the project team to do its work as they deem most appropriate as well as the ability to share people and their technical expertise across multiple projects when necessary. This strategy enables the organic self-organizing structure critical to the innovative approaches being implemented.

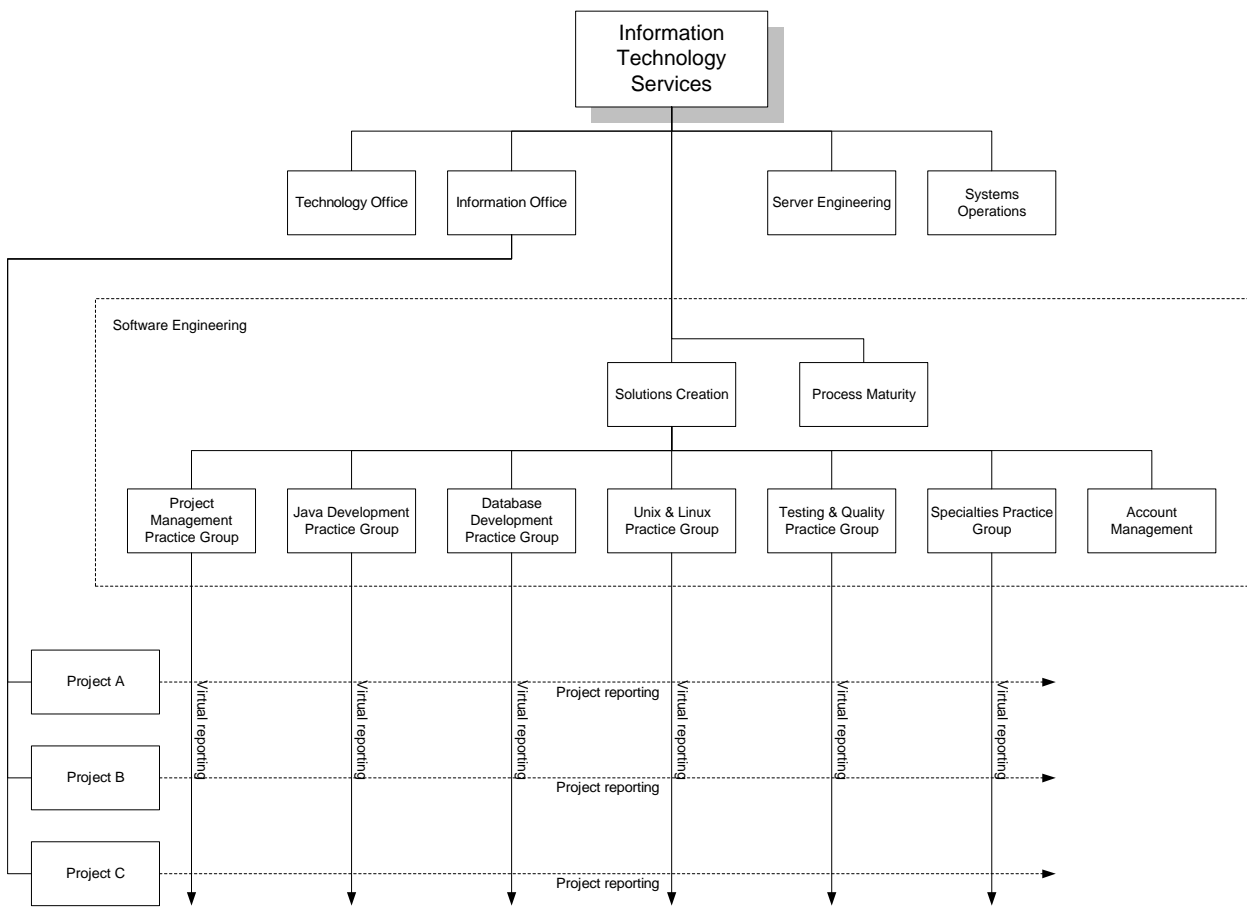


Figure 2 - Matrix Organization (adapted from Kerzner, 1998; p. 110)

This matrix organization is of primary significance because all work of business value to the customer is accomplished by project teams. Project team members have only virtual reporting relationships to the Software Engineering functional organization. These are virtual assignments in the sense that people are always assigned to project teams and their project manager holds all formal authority. The functional organization is very important, but is of secondary significance because it exists to provide people with necessary technical and administrative support that is outside the context of their active project assignments. Relative to the staff size for project work (about 300), the functional organization consists of very few full time staff (about 15 to 20). Table 3 describes the purpose as well as the nature of technical and administrative support functions of each component in the functional organization.

Table 3: Functional Organization Components

<i>Organization</i>	<i>Description</i>
Information Technology Services	Overall umbrella for the ITS function.
Technology Office	Interface to external technology environment. Researches and evaluates current and emerging technology for appropriate application to business need. Conducts product evaluations and assists project teams in product selection.
Information Office	Interface to customers in internal business organizations. Prepares business cases and assists business in prioritizing investment opportunities. Represents customer (end user) in liaison function by participating in project teams. Primary customer of Software Engineering organization.
Server & Network Engineering	Designs and implements hardware, network, and system software solutions in support of application projects. Assists System Operations in supporting production applications.
Systems Operations	Operates and supports production applications.
Process Maturity	Knowledge management function that mines best practices for processes from project work and promotes use across project teams.

Organizational Structure for Innovative Software Development

Organization	Description
Solutions Creation	People and skills management function for staffing application development projects. Component organizations represent a pool of people with specialized skills organized by practice area. Leaders within these components facilitate project team staffing, performance and salary review, skill inventories, career counseling, professional development, “communities of practice” (Daft, 2001; p. 265), cultural norms, and project assignment status as well as other administrative matters.
Project Management Practice Group	Focus on building core competencies for people who are or aspire to be practicing project managers.
Java Development Practice Group	Focus on building core competencies for people who are or aspire to be Java developers.
Database Development Practice Group	Focus on building core competencies for people who are or aspire to be database developers.
Unix and Linux Practice Group	Focus on building core competencies for people who are or aspire to be operating system technicians.
Testing and Quality Practice Group	Focus on building core competencies for people who are or aspire to be testers or quality assurance.
Contributing Specialties Practice Group	Focus on building core competencies for people who are or aspire to be practicing in other technology areas such as configuration management, human interface design, etc. .
Account Management	Focus on interfaces to Information Office and Human Resources providing liaison function for future project work and staffing needs.

Environmental Contingencies

Environmental contingencies are dependencies that exist between the Software Engineering organization and conditions external to it (Daft, 2002; p. 24). For the Software Engineering organization, these external environmental factors take two forms: those that exist outside of the Software Engineering organization between other organizational units within the company, and those that exist outside the company altogether. Table 4 identifies the domain of relevant external environmental factors (Daft, 2001; p. 131) and elements of the organization structure that perform a boundary spanning role (Daft, 2001; p. 141) or interface function for the Software Engineering organization.

Table 4: External Environment Factors

<i>Environmental Sector</i>	<i>Accommodating Organizational Structure Element</i>
Technology (tools and techniques, emerging technology applications)	Technology Office
Market (internal customer)	Information Office, Account Management
Human Resources (internal department)	Account Management
Financial Resources (project funding)	Information Office
Raw Materials (facilities, equipment)	Information Office
Economic Conditions (investment returns, growth rates)	Information Office
Government (regulations)	Information Office
Sociocultural (workforce)	Account Management, Practice Groups

Requisite Leader Characteristics

The change represented by implementing these new and innovative software development practices moves the organization toward becoming a learning organization. This kind of change is substantial in its impact on people, process, culture, and structure and its successful implementation requires a transformational leader (Daft, 2001; p. 507). Yukl (2002, p. 263) lists the guidelines for a transformational leader: “articulate a clear and compelling vision, explain how the vision can be attained, act confidently and optimistically, express confidence in followers, use dramatic, symbolic actions to emphasize key values, lead by example, and empower people to achieve the vision.” A leader exhibiting these characteristics and having demonstrated experience in applying them is a critical success factor.

References

Daft, R. (2001). Organization Theory and Design. (7th ed.). Cincinnati: South-Western College Publishing.

Fowler, M. (2002). The New Methodology. Retrieved from the World Wide Web September 26, 2002: <http://www.martinfowler.com/articles/newMethodology.html>

Jeffries, R. (2001). What is Extreme Programming? Retrieved from the World Wide Web September 26, 2002: <http://www.xprogramming.com/xpmag/whatisxp.htm#design>

Kerzner, H. (1998). Project Management – A Systems Approach to Planning, Scheduling, and Controlling. (6th ed.). New York: Wiley & Sons.

Senge, P. (1998). The Practice of Innovation. Leader to Leader, 9 (Summer). Retrieved from the World Wide Web September 24, 2002:
<http://www.pfdf.org/leaderbooks/l2l/summer98/senge.html>

The Standish Group Report. (1995). Retrieved September 24, 2002 from
<http://www.scs.carleton.ca/~beau/PM/Standish-Report.html>

Yukl, G. (2002). Leadership in Organizations. (5th ed.). Upper Saddle River: Prentice-Hall