

***Organizational Structure for Innovative Software Development in a  
Large Corporation***

*Joseph C. Thomas*

*Regent University School of Leadership Studies*

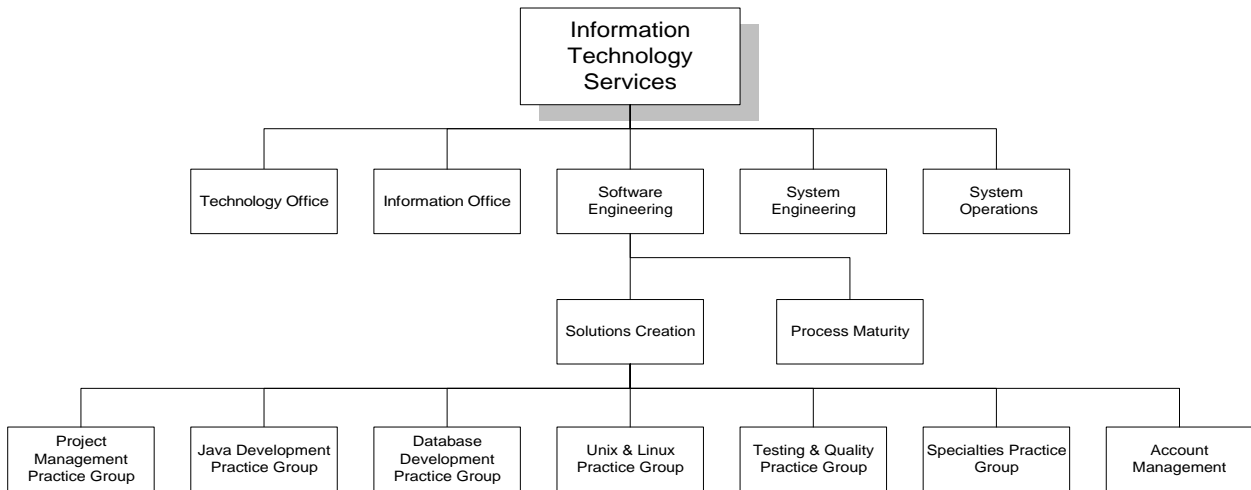
*LEAD606 Strategic Vision and Organizational Effectiveness*

*9-Dec-2002*

## **Introduction**

This paper concludes a three part presentation that describes a software development organization in a large corporation. The first paper presents the mission and purpose, organizational structure, environmental contingencies, and requisite leader characteristics. The second paper presents a refined vision statement.

This paper further elaborates on elements in the first paper including environmental contingencies and the relationship of leader characteristics to the organization's mission, structure, and environment. This paper also describes additional considerations including employee structure, organizational communication, recruiting and training, vision communication, building a learning culture, and changing the organizational culture.



**Exhibit 1 -- The ITS Organization**

## **Environmental Contingencies**

The Software Engineering organization faces several significant challenges that center around the market, technology, and human resource sectors of the external environment

as described by Daft (2001, p. 131). Elements in the organizational structure address these environmental challenges.

In the market sector, the challenge is to build credibility and trust with the Information Office (Software Engineering's primary customer). Currently, the Information Office organizes software development projects by appointing project managers, and considers the Software Engineering organization as a source of personnel with which to staff project teams. This approach causes projects to be disjoint efforts that fail to fully leverage investments in people, processes, and technology. Account Management addresses this challenge by nurturing the relationship between Software Engineering and the Information Office through actively understanding and managing current and future technology project needs. This facilitates the migration of project organization and staffing functions from the Information Office to Software Engineering. Organizing and staffing projects positions Software Engineering for productivity improvements and provides the opportunity to build customer confidence in the ability to deliver solutions.

Also in the market sector, challenges of a fast paced business environment with uncertain, changing, and difficult to articulate requirements is the norm. Project teams as an organizational structure component provide focus on core business processes of delivering software solutions and enable sharing of technical staff across multiple efforts (Daft, 2001, p. 104). Organizing around project teams also allows simultaneous development of multiple solutions, and facilitates interaction with a complex and uncertain environment (Daft, 2001, p. 104).

In the technology sector, the challenge is to selectively employ rapidly changing computer technologies and innovative methods and tools as they mature and become relevant to business needs. The Technology Office addresses this challenge by researching and evaluating current and emerging technology for appropriate application to business need, evaluating products, assisting project teams in product selection, and providing advance information for use by Practice Groups. This allows timely application of new technology to project needs, deployment of innovative methods and tools, and propagates technical knowledge amongst technical staff.

In the human resource sector, the challenge is to attract and retain qualified information technology professionals. Currently, hiring of new staff is driven by headcount with little emphasis on meeting the skill needs of projects. This approach causes project staffing skill and personnel shortages, inadequate staff utilization, and staff turnover. The Practice Group organizations, working with Account Managers to understand customer needs, address this challenge by focusing on acquisition and retention of professional staff with the right skills. Practice Groups realign Software Engineering staff into technology skill groups and focus on professional development of staff and creating a sense of community. This allows more effective matching of project needs and people skills, facilitates retention of qualified staff, and enables more effective team work.

### ***Characteristics and Role of Leadership***

The leader of Software Engineering must possess essential skills common to all leaders such as the ability to effectively communicate and establish vision and mission. Given the substantial challenges of transforming an existing large and primarily bureaucratic

technical organization, a transformational leader successful in affecting organizational and cultural change is necessary. Subsequent sections of this paper describe the nature of communicating the vision, building a learning culture, and affecting cultural change. There are other requisite leadership skills specific to Software Engineering's environment, mission, organizational structure, and staff.

Leadership skills in the environment area include coalition building within the organization and formation of alliances with outside vendors. To foster cultural change and build trust, the leader must be able to relate effectively with Information Office leadership and staff. The leader must persuade superiors, peers, and staff to adopt the vision and participate in its implementation. The leader must also form alliances with outside staffing firms to ensure a steady flow of qualified contractors at cost-effective rates.

Leadership skills in the mission area include technical expertise and knowledge management. The leader must have experience with the strategic technologies of the company and take an active role in guiding development and implementation of advanced software development and project management methodologies. This is necessary because the leader must be technically credible to build trust with the technical staff and leadership peers. The leader must also establish an effective knowledge management function to discover and propagate best technical practices across the Software Engineering organization.

Leadership skills in the organizational structure area include organizational design concepts (for example, bureaucratic and learning organizations) and the ability to lead a

change management program to implement the new organizational structure. The leader must make significant changes in organizational structure by realigning the technical staff to form Practice Groups and reorient the formation of project teams. The leader must also quickly establish a leadership team that can carry forth the new vision and implement change throughout the organization.

Leadership skills in the staff area include relevant technology experience and building of trust with staff. The leader must build credibility with the technical staff by demonstrating expertise with applicable technologies (innovative software development and project management methodologies, Java development, n-tier application design, internet, etc.). The leader must also build trust amongst the staff because they are jaded by repeated failure to bring about meaningful change in the management-staff relationship.

### ***Employee Structure and Organizational Communication***

The Software Engineering organization uses a hybrid structure combining hierarchical and project teams in a dual authority matrix organization. Hierarchical and team structures are compatible and complementary (Avery, 2000, p. 4), and this type of organizational structure is appropriate when technical expertise, innovation, and change are important to achieve business objectives (Daft, 2001, p. 103).

The hierarchical accountability structure formed by the Practice Groups organizes employees into groups by skill or role which is effective for staffing project teams (Avery, 2000, p. 4). The Practice Group organizations also provide for handling administrative

matters such as annual performance reviews, rewards (promotions, bonuses, and salaries), expenses, time reporting, and lateral moves between skill communities.

Practice Groups also provide a vertical communication mechanism (linkage) between employees with common technical interests (Daft, 2001, p. 88). This is important for employees, who are almost always assigned to project teams, to feel connected to the larger Information Technology organization. Practice Group managers are responsible for enabling the building of communities of practice where employees with common skill sets come together to share project experiences, help one another with situational problems, and promote industry knowledge. Additionally, communication by communities of practice is facilitated by the use of software collaboration tools such as news groups, software asset reuse (to catalog software, retrieve code, and serve as a reference tool), and a software development environment (to manage code, content, and process as well as to track, measure and report on software project activity).

The project team structure organizes employees into teams that focus on the core processes of developing and delivering software solutions, with project managers (process owners) having direct responsibility for delivery of value to the customer (Daft, 2001, p. 109). Such team structures eliminate departmental barriers that would otherwise inhibit communication between the Practice Groups (Daft, 2001, p. 89, 92).

### ***Recruiting and Training***

The recruiting strategy consists of developing a close relationship with a sole source technical contracting vendor, working with the vendor to develop an effective candidate

screening process, bringing in technical staff as contract employees for specific project assignments, and identifying the most promising contractors and converting them to company employees. This approach seeks to ensure hiring only the best technical staff.

Staff retention is more of a critical success factor than recruiting. Currently, employees feel little affinity toward their staff groups and peers in other projects and little connection with the Information Technology organization and its higher level goals. This causes poor morale and staff turnover especially when projects draw to a close as employees feel connected primarily to their project teams. Practice Groups provide the opportunity for employees to connect with peers, other project teams, and the overall organization.

Information Technology will continue to invest in technical skills training for staff but knowledge management is the basis of the training strategy. Knowledge management is “organizing and sharing an organization’s intellectual and creative resources” using “efforts to systematically find, organize, and make [them] available ... to foster a culture of continuous learning and knowledge sharing” (Daft, 2001, p. 257). Two knowledge management approaches comprise the training strategy: knowledge mapping and communities of practice.

Knowledge mapping is discovering best practices within the organization and making them available for use by others (Daft, 2001, p. 262). The Process Maturity group is responsible for mining best practices from individual projects, disseminating them across all project teams, and working to incrementally improve them in real project situations.

Communities of practice are groups of people with common interests that coalesce to deal with issues and pursue solutions (Daft, 2001, p. 265). Although communities of practice are informal and they cannot be directly managed (Daft, 2001, p. 265), the Practice Group organizations within Software Engineering are organized around technology areas or skills, and Practice Group managers are charged with providing opportunities for communities of practice to form and grow. Various kinds of employee gatherings such as celebrations, informal lunch time training sessions as well as bringing in industry and company speakers provide these opportunities.

### ***Vision Communication by Leadership***

Communicating the vision is critical to affect organizational change and to get people to support the change (Yukl, 2002, p. 283). For Software Engineering, there are two audiences for communicating the vision, each with its own strategy: customers within the Information Office and Software Engineering staff. It is critical for the leader of Software Engineering to take an active and visible role in conducting these communication programs to demonstrate commitment and lead by example.

For the Information Office, the goal is to gain buy in to changing the organizational structure so that Software Engineering is in better position to provide turnkey project teams that address the application software needs of the business. The communication program includes a seminar series to promote the vision with an emphasis on developing the Account Manager role and Project Manager Practice Group, development of individual action plans that target key Information Office leaders and staff, and identifying

opportunities to speak at Information Office events. Regular review of progress and adjustments as a result of lessons learned are part of the plan.

For Software Engineering, the goal is to gain buy in to the core values and reorganization of the software development staff into Practice Groups. The communication program includes a seminar series to describe vision and explain organizational structure changes, initiation of focus groups to review recommendations for the number and type of Practice Groups, actively seeking opportunities to engage individual staff members at all levels in conversation about the vision, and publication of the vision statement in various forms (brochures, articles in company newsletters, intranet, etc.). Regular review of progress and adjustments as a result of lessons learned are part of the plan.

### ***Building a Learning Culture***

A learning culture increases the capacity of the organization to solve problems and improve capability by engaging everyone in continuous communication, experimentation, and collaboration (Daft, 2001, p. 25). Kofman and Senge (1993, p. 2) assert that building a learning community requires personal commitment on the part of each individual and a “fundamental shift in how people think and interact with one another,” describing this as genuinely caring for one another, taking on an air of humility, exhibiting authenticity in personal relations, demonstrating servant leadership, and taking risks through practical experimentation. The leader of Software Engineering builds a learning culture by taking an active role in establishing these values, teaching others about them (especially within the leadership team), initiating conversations at every opportunity that reinforce these values, and leading by example.

To complement these activities, the leader teaches and models double-loop learning as a healthy means to deal with organizational problems and interpersonal conflict. Double-loop learning “corrects errors by examining the underlying values” and organizational policies that cause problems (Argyris, 1993, p. 2). The most difficult problems in a group revolve around interpersonal conflicts between team members, and defensive behaviors in these situations can prevent meaningful action and sustainable resolution (Argyris, 1993, p. 4). The leader of Software Engineering identifies these defensive behaviors, and models alternative reasoning and corresponding action based upon objective interpretation of conversations and events (Argyris, 1993, p. 2, 4).

### ***Building Organizational Culture***

Culture consists of the values, attitudes, expectations, beliefs, abilities, and behavior of the people that comprise the organization (Daft, 2001, p. 357). Key aspects of the culture to which Software Engineering aspires are represented by the Guiding Philosophy or core principles (People Principles, Process Principles, and Technology Principles) contained within the Vision Statement. Previous sections of this paper have also discussed characteristics of the leader, vision communication, and building a learning culture, and these discussions all address aspects of building organizational culture.

Although these activities are important aspects of influencing culture, they have little credibility unless they are backed up by leader decisions and actions in the context of day to day activities (Yukl, 2002, p. 281). Accordingly, the leader must be cognizant of several primary ways to influence culture including attention, reaction to crises, and role

modeling (Yukl, 2002, p. 279-280). Attention refers to what the leader spends time talking about, planning, measuring, praising, and criticizing. Reaction to crises refers to the leader's often emotional and spontaneous response under pressure to unexpected events. Role modeling refers to the actions a leader takes as they go about day to day activities. What garners the attention of the leader, how the leader handles crises, and their routine activities all communicate priorities and values.

Allocation of rewards and the criteria employed for recruiting, hiring, and dismissing people are two other primary methods of influencing culture that the leader can employ (Yukl, 2002, p. 280). Accordingly, the Software Engineering leader must carefully consider policies for these two activities and implement appropriate programs.

### ***Conclusion***

A good vision statement is foundational for a leader to transform and build an organization, but it is only one aspect of ensuring the ongoing success of an organization. Other critical facets include the organizational structure, environmental contingencies, leader characteristics and role, communication, employee structure, recruiting and training, and organizational culture. All of these aspects of organizational theory collectively form a framework that a leader must align with the overall strategy of the organization to achieve the creation of a dynamic transforming organization.

## **References**

Argyris, C. (1993). Education for leading-learning. Organizational Dynamics 21, (2) (Winter). 5-18.

Avery, C. (2000). How teamwork can be developed as an individual skill. The Journal for Quality and Participation (23), (4) (Fall). 6-13.

Collins, J. & Porras, J. (1991). Organizational vision and visionary organizations. California Management Review 34, (1). 30-53.

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

Kofman, F. & Senge, P. (1993). Communities of commitment: the heart of learning organizations. Organizational Dynamics 22, (3) (Autumn). 5-24.

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