Software Executive Report

Organizational Structures

Editor's Note: I've commented before that, while "organizational structure" is rarely discussed, we have found it to be critical. Organizational issues apply at the corporate level, department level, and project team level. Over the years we have found an amazing variety of approaches used by different companies. This report describes some of those variations.

— *Steve McConnell*

Functional vs. Product Organizations

At the top level, the most common organizational issue is whether to organize the software staff along functional lines (dev, test, project management) or product lines.

Advantages cited for functional organizations include efficiency, better ability to learn best practices/establish centers of excellence, and better ability for functional managers to manage people whose skill sets they understand.

Disadvantages of functional organizations include staff feeling less connected to products they're working on, the promised benefits of staff development not materializing, and various reporting conflicts—e.g., the classic "I have five bosses who all think they get 100% of me" syndrome.

Advantages cited for product organizations include staff identifying more strongly with their company's products, and staff being more closely connected with product completion and achievements.

Disadvantages of product organizations include a lack of focus on skills development/ career development and staff having difficulty rotating to a different product area after they have become established in one product area.

In light of the inconclusive strengths and weaknesses of these different organizations, companies report numerous philosophies that drive their decisions about functional vs. product organization:

◆ "Product line vs. Functional organization is a case of what's good for the company vs. what's good for the staff."

◆ "Functional vs. product organization has to do with the maturity of the business— early in the business lifecycle you need functional people together so they can learn from each other."

◆ "A product organization can work well when a product has a well-defined market as a standalone product."

◆ "Functional can be a good fit when the team or organization is big enough to realize economies of scale."

◆ "Manufacturing companies often have functional organizations (to support

February ECSE Meetings
Managing Global Development

Companies pursue global development for many reasons including access to specialized talent, lower labor costs, access to talent with better market understanding of local markets, and general geographic diversification. This discussion will focus on aspects of software development that change when development goes global— management issues, technical issues, warning signs, technical keys to success, and management keys to success.

Bellevue Meeting: February 11, 5:00-7:00 pm. **Dial-in Meeting:** February 15, 8:00-9:00 am, Pacific Time.

Construx®
www.construx.com

Construx®

(Continued from page 1)

manufacturing which is usually functional in structure)."

Advantages to Changing Strategy Periodically

Executives sometimes wonder why reorganizations happen so frequently—why there seems to be a pendulum swing in many organizations from functional to project back to functional back to project.

Most executives report reorging every few years—swinging the pendulum from product organization to functional organization and back again. Most executives report that reorging in itself is healthy—it clears out organizational cobwebs, creates new opportunities for staff members who have gotten pigeonholed, creates an opportunity to clean out dead wood, provides more opportunity to gain career-development experiences, and so on.

> *"Most executives report that reorging is healthy in itself."*

Organization Within Projects

Lots of companies in the Pacific Northwest seem to use the Microsoft model of program manager, project manager, dev lead, QA lead, developers and testers.

Another common (related) organization is feature crews or feature teams in which developers, testers, and managers all work together in a team. **Because all project stakeholders are represented within the team, the team can collectively own decision making about the schedule, feature set, and most other decisions required for delivery.**

Regardless of organizational details, most companies have found that keeping team sizes to no larger than 8-10 people makes them most manageable.

Management Responsibilities

Organizations report having various flavors of product managers, program managers, and pro-

ECSE Calendar 2008	
February	Managing Global Development
March	Navigating the Planning Cycle
April	Managing Core Development
May	Special Issues in Managing Technical Personnel (aka "Guru Management")
June	Balancing "Doing" with "Improving": Improvement Strategies
July	Supporting Innovation
August	Summer break
September	Issues in Test Management
October	Compensation Updates
November	Improving Productivity
December	*To be announced*

ject managers—with little consistency in how these terms are used. Most organizations have nominal job descriptions for each of these "p-manager" titles.

◆ **Product managers** are not deeply technical but understand their product and are responsible for eliciting requirements and working out conflicts in the product definition. This can be a strongly customer-facing role and is often organized as part of marketing rather than as part of development.

◆ **Project managers** are usually strongly technical and are responsible for managing resources and schedule. They are usually organized as part of development. The title of dev manager is usually used more or less interchangeably with the title of project manager.

◆ **Program managers** are responsible for meeting the business case—conceptually the product manager is marketing, the project manager is dev, and the program manager is general management.

On small projects, these roles are often combined in individuals—the project manager may also be the program manager, or the product manager may be the program manager. A large

(Continued from page 2)

project may have multiple product manager and project managers but normally has only one program manager.

In practice, the strengths and weaknesses of the specific individual in a p-manager role usually exerts as much influence on the specific job definition as the general job description does.

Organization of Dev vs. Test

Companies grapple with dev vs. test organizational test issues including reporting chain and ratios of developers and testers. In these areas, a few best practices seem to have emerged.

Having test located within each project team seems to be a clear good practice (or even best practice). Many companies have reported trying to separate dev and test along geographic lines (i.e., using offshore org for testing and doing dev onshore). In nearly all cases companies report abandoning such organization in favor of collocating all contributors who are working on the same component. There is near universal consensus that physical proximity of dev and test is beneficial.

A key issue is having development retain ownership of the quality of the code they produce, and the details of the dev/test organization should support that. As one executive said, "To get a quality product out of test you have to put a quality product into test." **Test's job is to make sure that the quality level is known; not to put quality into the product.** Companies report that developers finding 65-80% of their own defects is normally seen as very good.

> *"Most companies have found that keeping team sizes to no larger than 8-10 people makes them most manageable."*

Dev and Test Ratios

The ratio of dev: test depends both on the nature of the software being developed and on the specific development approach used by the organization. In very general terms, for high reliability software ratios ranging from 2 developers to tester to 0.5 developer per tester are commonly reported. For very broad distribution software (shrink wrap), 1:1 ratio is common. For vertical market shrink wrap, ratios of 2:1 to 4:1 are common. For general business software, ratios of 3-10 developers per tester are common.

Dev and Test Reporting Structures

In years past having separate dev and test reporting structures—often not converging until the functional VP level—was seen as good practice. **In more recent years that organization has been found problematic, leading to issues including test not feeling ownership for product releases and the dev/test relationship becoming adversarial.**

The most common management structure is that developers report at the project manager level while testers report to a functional QA or test manager. Both dev and test managers then report to the same Product Director or VP.

About the Executive Council for Software Excellence (ECSE)

The ECSE is an executive discussion group hosted by Construx Software. Meeting monthly since 2002, the ECSE's goal is to share, analyze, and evaluate members' experiences facing enterprise-level software development challenges. ECSE members are executives with multi-project, enterprise-level responsibility for software development. The typical member oversees activities of 100 or more software personnel. The ECSE has a few members who oversee smaller staffs.

If you are interested in joining the ECSE or if you know someone who would be interested, please contact the ECSE host, Steve McConnell, at **stevemcc@construx.com** or (866) 296-6300.
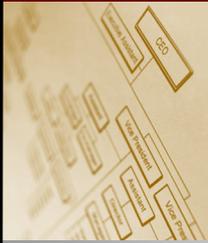
The exception is companies who work under FAA or FDA regulations, which usually want to see separate (aka independent) QA organizations. In these organizations the QA or V&V reporting structures might not meet until much higher. In one company, these reporting structures don't meet until the COO.

Impact of Multi-Site Development on Organizational Structures

As described earlier in this report, many companies report having migrated away from doing test in one location and development in another—preferring to co-locate teams working on the same software.

A few companies report that doing multi-product integration testing offshore has worked well even with development being kept onshore. However one company reported that offshoring compatibility testing to China was a case study in how not to do offshore work – problems included not being able to get hardware that you can easily get in the U.S., communications problems, and so on.

One company reported an emerging consensus that **you need a critical mass of 30-50 people at one site to work on a technology together**, so most groups try to compartmentalize the offshore group ("project in a box") to reduce the communication challenges. Many companies have their entire offshore organization report into their development org.

Challenges include:

◆ The time difference with India wears after awhile.

◆ It's hard to keep the time inconvenience evenly distributed across the sites, meaning

> *"Having test located within each project team seems to be a clear good practice (or even best practice)."*

that usually the offshore site bears the brunt of the time inconvenience.

◆ **"Round the clock development" can turn into "Two day development" when a question that was left overnight doesn't quite get answered the next day and has to be re-asked and re-answered.**

◆ There are difficulties in training and strategic planning with offshore.

◆ One large company reported an immediate 25% drop in productivity when it went offshore.

Communication between onshore and offshore staff remains a challenge. Companies address this challenge in numerous ways:

◆ One company uses "on shore coordinators" in India who report to project managers in the US.

◆ Working with captive orgs is more effective than working with independent ones.

◆ Face-to-face meetings are critical. Travel will be required.

We'll discuss issues related to global development more in our next issue. ■

Construx and Organizational Consulting

Construx provides numerous services that can help identify organizational issues and make improvements. Our services include Organizational Assessments, Organizational Benchmarking/ Report Cards, SDLC definition and reviews, and other specific services created to meet the needs of individual clients. For information about how Construx can help your global organization, please contact us at consulting@construx.com or +1 (866) 296-6300.