T. H. E. SOLUTION
Product Development Consulting

PRACTICE LEVELS AND ASSOCIATED BENCHMARKS

Changing the maturity level of your technical development practices can greatly affect your organization's performance. A meaningful discussion of this topic requires the establishment of a basic framework. Defining three maturity levels, basic, good and best, provides the framework (Note: A bad level is implied as less than basic maturity). This is artificial since real organizations fit a continuum from bad to best but it does provide a means to keep the comparisons tractable. According to the Software Engineering Institute most software projects (including those embedded in complex product developments) would not meet even the basic level of maturity.

The basic level of practice is based on the adaptation of an organized approach (a corporate level process model has evolved, well known top-down engineering principles have been put into place and these are supported by at least a minimal tool set). This state can normally be achieved without a large expenditure of money but it does require a significant amount of effort. A summary of basic level practices for three key technical disciplines, project management, software engineering and quality assurance follows:

Discipline	State of Practice
Project management	process phases and work products defined formal review between phases organization/role definitions based on multi-functional teams project metrics selected but little historical data project status parameters loosely monitored over commitment of resources recognized and corrected basic group work tools in place
Software engineering	process steps and work products defined automated configuration management primary metrics in place but little historical data peer reviews at least between steps system integration testing before release to evaluation source closely scrutinized for language violation practices normally followed
Quality assurance	inspection/testing aimed at unit, sub-system and system level formal verification in system level testing automated defect tracking and some automation of testing some QA personnel participation in early work product review

In a good level of practice the organization has increased the rigor used during development and

tightened relationships between parts of the company and with the customer(s). This requires a significant investment in people, process and tools. A summary of improvements that result in a good level of practice for each of the three disciplines follows:

Discipline	State of Practice
Project management	regularly learn from past tight integration with marketing, operations AND customer roles well understood and comfortable planning complete and sanity checked with historical data project status parameters closely monitored over commitment of resources not allowed
Software engineering	formal inspections of all work products high automation of design and documentation source scrutinized for any sign of potential problems practices consistently followed all participants trained in process and tools
Quality assurance	formal reliability engineering approach in place most testing automated heavy QA personnel participation throughout

The best practices are extensions of good practices. They are primarily driven by a skilled work force with extensive experience in a specific environment and the maximum reuse of proven good "components". Since these conditions normally cannot exist for all development projects, it is not realistic to expect to sustain this level of maturity across all developments. A summary of changes that result in the best level of practice for each of the three disciplines follows:

Discipline	State of Practice
Project management	majority of work force is experienced in process strong emphasis on risk management
Software engineering	majority of work force is experienced in process and tools good domain knowledge low performers have been eliminated heavy reuse of existing proven correct code
Quality assurance	defect cause information used in root-cause analysis

There is a significant body of information on the improvements made by some more forwarding thinking organizations (e.g. parts of Raytheon, AT&T and IBM) during the late 80's and early 90's. It indicates that moving from basic to best practice can result in a fivefold decrease in project schedule and cost variability, an order of magnitude increase in software productivity and two orders of magnitude increase in software quality. A summary of some key benchmarks

follows:

Project Performance Benchmarks:

Level of Practice	Cost of quality (% sales)	Schedule (max % over plan)	Development cost (max % over plan)
basic	10	50	50
good	5	20	20
best	3	10	10

cost of quality - the amount expended to meet quality objectives (e.g. scrap, inspection, quality assurance etc.)
schedule - the duration of a development project
development cost - the total cost of a development project to include effort, materials and expenses

Software Productivity and Quality Benchmarks:

Level of Discipline Practice	Productivity (LOC/man-month)	Quality to Customer (failures/KLOC)
basic	250 - 600	4
good	700 - 1500	.1
best	> 2000	.001

LOC - lines of code
KLOC - thousand lines of code
failures - deviations from requirement/specification during execution

Obviously, there are real benefits to be gained by improving your technical development practices. But, as in home construction, an imposing edifice built on sand may look grand but it will not stand the test of time. In other words, the effective and efficient development of the wrong product is still a business failure. If corporate behavior does not promote customer satisfaction while meeting company objectives, long term profitability etc., there is no amount of change to the technical development organization that will make any significant difference to the business outcome. If, however, the basic corporate framework is securely in place, then improving technical development practices can serve as a competitive advantage that fosters long-term profitability.