Capers Jones, Software Productivity Research Inc., 1 New England Executive Park, Burlington, MA 01803-5005; phone (617) 273-0140; fax (617) 273-5176; Internet capers@spr.com; CompuServe 75430,231

The economics of software process improvement

Capers Jones
Software Productivity Research

Software process improvement is gaining momentum throughout the software industry. Many cities now have nonprofit SPIN (Software Process Improvement Network) groups. In a visible sign that process improvement is now a mainstream technology, SPIN's national conference in Boston last spring drew several thousand attendees. Another sign is the frequency of journal articles devoted to process improvement.

However, as often happens with software, process improvement articles tend to be theoretical. Comparatively little solid, empirical data is being published on three important topics:

- What does it cost to improve software processes?
- How long will it take to make tangible improvements?
- What kind of value can be expected in terms of better quality, productivity, or user satisfaction?

Both process assessments and baselines are necessary. Assessments alone do not provide initial quality and productivity levels, which are needed to judge improvements.

My latest book, *Patterns of Software Systems Failure and Success* (International Thomson Computer Press, 1995), addresses the economics of software process improvement for companies of various sizes for five different sectors: systems software, military software, information systems, outsource vendors, and commercial software. This column condenses some of my findings, which are based on studies of leading software producers in the United States, Europe, South America, and the Pacific Rim.

Optimal sequence

Significant software improvements do not occur at random. By generalizing the patterns used by the most successful companies, we see that the initial activity is an assessment and a baseline, followed by a six-stage improvement program.

Stage 0—Process assessment and baseline. All software process improvement strategies begin with a formal process assessment and a quantitative baseline of current productivity and quality levels. The assessment is like a medical diagnosis of the organization's software strengths and weaknesses. The baseline provides a firm, quantitative basis for productivity, schedules, costs, quality, and user satisfaction. Both are necessary, since assessments alone lack the quantification of initial quality and productivity levels needed to judge later improvements.

Stage 1—Management technologies. Because management is the weakest link in the software process, the first improvement stage concentrates on bringing managers up to speed in critical technologies such as planning, sizing, estimating, tracking, measurement, risk analysis, and value analysis. Managers are the ones who must calculate the return on investment (ROI) and collect the data to demonstrate progress. Accordingly, it is folly to begin improvements unless managers are trained and equipped for their roles.

Stage 2—Processes and methodologies. The second stage concentrates on solid approaches for dealing with requirements, design, development, and quality control. Since tools support processes, the processes must be selected before heavy investments occur. Some processes deployed in this stage include joint application design (JAD); formal design methods such as Warnier-Orr or Yourdon; formal design and code inspections; and formal change-management procedures.

Stage 3—New tools and approaches. Once methods and processes are pinned down, it is appropriate to acquire improved tools and explore new technologies. This is the time to acquire integrated CASE (I-CASE) and other new or upgraded tools. This is also the time to explore difficult technologies with steep learning curves, such as client-server methods and the object-oriented paradigm. Jumping prematurely into client-server projects, or moving too quickly toward object-oriented analysis and design,

Table 1. Process improvement costs per employee by size of software staff.

Stage	Small staff (<100)	Medium staff (101–1,000)	Large staff (1,001–10,000)	Giant staff (>10,000)
Stage 0—Assessment/baseline	$100	$125	$150	$200
Stage 1—Management	$1,500	$5,000	$5,000	$8,000
Stage 2—Methods/processes	$1,500	$2,500	$3,000	$3,500
Stage 3—New tools	$5,000	$7,500	$15,000	$25,000
Stage 4—Infrastructure	$1,000	$1,500	$3,500	$5,000
Stage 5—Reusability	$500	$3,000	$6,000	$7,500
Stage 6—Industry leadership	$1,500	$2,500	$3,000	$4,000
Approximate total	$11,100	$22,125	$35,650	$53,200

usually means trouble because poorly trained practitioners are seldom successful.

Stage 4—Infrastructure and specialization. Achieving software excellence requires top-notch organization as well as excellent tools and methods. The infrastructure phase addresses organization and specialization issues and moves toward establishing specialized teams for handling testing, maintenance, integration, and configuration control. Formal quality assurance departments are also important, as are policies on continuing education.

Stage 5—Focus on reusability. Reusability has the best ROI of any technology, but effective reuse is not a game for amateurs. If a company's quality control is poor, it will be reusing garbage, and its costs will go up instead of down. Also, effective reuse includes much more than just code. In fact, an effective software reuse program includes 10 reusable components: reusable architecture, requirements, plans, estimates, design, interfaces, data, source code, user documents, and test materials.

Stage 6—Industry leadership. Organizations that reach the sixth stage are invariably leaders in their indus-

Table 2. Months for each improvement stage by size of software staff.

Stage	Small staff (<100)	Medium staff (101–1,000)	Large staff (1,001–10,000)	Giant staff (>10,000)
Stage 0—Assessment/baseline	2	2	3	4
Stage 1—Management	3	6	9	12
Stage 2—Methods/processes	4	6	9	15
Stage 3—New tools	4	6	9	12
Stage 4—Infrastructure	3	4	6	9
Stage 5—Reusability	4	6	12	12
Stage 6—Industry leadership	6	8	9	12
Approximate total	26	38	57	76

Table 3. Improvements in software defect levels, productivity, and schedules.

Stage	Defect level	Productivity gain	Schedule reduction
Stage 0—Assessment/baseline	—	—	—
Stage 1—Management	10%	—	10%
Stage 2—Methods/processess	–50%	25%	15%
Stage 3—New tools	10%	35%	15%
Stage 4—Infrastructure	5%	10%	5%
Stage 5—Reusability	85%	65%	50%
Stage 6—Industry leadership	5%	5%	5%
Approximate compound totals	90%	350%	70%

tries. These organizations are often in a position to acquire competitors or become outsourcers.

Costs, timing, and value

For accuracy, each company must create an individualized plan and budget for its improvement strategy. Table 1 presents general information based on company size in terms of software personnel. The cost data, expressed in terms of approximate cost per software employee, includes training, consulting fees, capital equipment, software licenses, and office improvements.

An interesting question is, what is the smallest expenditure that will achieve any tangible benefits? The cheapest improvement strategies appear to be methodological improvements that do not require capital investments in new tools. Joint Application Design (JAD), function-point metrics, and inspections, for example, can be deployed with only training expenses.

Two-day training sessions for any one of the three above methods are available for about $800 per person, plus travel expenses. Since introducing any one of the methods produces tangible benefits, the minimum cost of visible improvements is roughly $800 per person. Such minimal investments, however, seldom result in more than a 10 percent improvement in schedules, costs, and quality.

Table 2 shows how long it takes to move through each stage of the process improvement sequence. Clearly, smaller companies can move more rapidly than large corporations and government agencies.

What kind of value or ROI can we expect from software process improvements? My experience in developing models and analyzing client data indicates an ROI, measured over a 48-month period, of $3.00 to more than $30.00 for every dollar invested in software improvement programs. Of course, not every technology will return its maximum ROI in every case.

Table 3 shows approximate improvements in schedules, costs, and quality (defined here as software defect levels). The results are expressed as percentages of the initial baseline.

A typical software application of 1,000 function points (equal to roughly 128,000 C source statements) should achieve the following results. Delivered defect levels should drop from about 1 per function point to 0.1 per function point. Software productivity rates should rise from about 8 function points per staff month to 28 function points per staff month. Development schedules should shrink from a nominal average of 18 months to 5.5 months. The volume of reusable material should rise from less than 15 percent to more than 65 percent for source code, specifications, test cases, and other key items.

From this rough analysis, we see that maximum benefits do not occur until Stage 5, when full software reusability programs are implemented. Since reusability has the best return, I'm frequently asked why it isn't the first stage. The reason software reuse is offset to Stage 5 is that a successful reusability program depends on mastery of software quality and a host of precursor software technologies such as inspections. Attempts to reuse materials not at state-of-the-art quality levels actually result in longer schedules and higher costs.

New Products

Contact or send releases to Scott Hamilton, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; Internet, s.hamilton@computer.org

Hardware

Embedded mouse solution

Usar Systems introduced the HulaPoint pointing device that combines the functionality of a mouse with the ease-of-use of a joystick. The device uses magnetic-field-detection sensor technology pioneered by Fujitsu to output coordinate data according to the direction and degree of travel of the mobile section. It offers 10-degree motion in all directions for positive cursor control and user feedback. The controller IC translates the sensor motion and provides autoselectable RS-232 and PS/2 output.

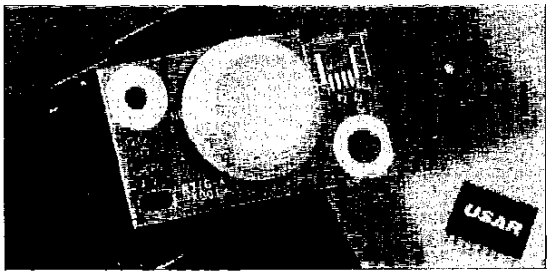
The device supports the IBM/MS two-button and Logitech three-button protocols. The mouse driver offers additional functionality and customizability that lets users optimize data input and screen operations. Users can control acceleration, command the pointer to grow while moving, and define hot spots to which the cursor will move upon command. An evaluation kit comprising the assembly, cables, and documentation costs $95.

Reader Service Number 36

Computer security

Jetico Inc. introduced BestCrypt+ Data Protection System for IBM-PC compatible computers. The hardware/software package lets users keep sensitive data encrypted on the disk and later access it transparently with any application program.

The PC-standard add-on board features a single-chip encryption processor implementing the 256-bit encryption method known as Russian federal standard GOST 28147-89. The software provides encryption services usable with both hard disks and floppies. Virtual disks with specific passwords are created on the hard disk. All information kept on the virtual drive is automatically decrypted during the read operations and encrypted on the write. Any application program can use the files stored on the virtual disk provided the user has the password. All operations are made transparently without slowing down the application, according to the company.



Usar Systems' HulaPoint combines the functionality of a mouse with the ease-of-use of a joystick.