

# Instilling Professionalism in a Software Development Organization

Conrad Weisert  
Information Disciplines, Inc., Chicago

from *Proceedings<sup>1</sup> of the 1988 ACM SIGCPR Conference on the Management of Information Systems Personnel*, April 7-8, College Park, Maryland

## Introduction

Whether data-processing can be considered a profession in the traditional sense is debatable. We can nevertheless talk about a professional approach to data-processing activities: a cohesive collection of the skills, roles, attitudes, support facilities, and organization structures aimed at attaining very high levels of quality, productivity, and manageability in application software development. Its success depends on a partnership between qualified staff members and an enlightened organization.

Such an approach is in sharp contrast with typical practices and attitudes found in most data-processing organizations. Furthermore, it's being threatened by recent trends, such as uncontrolled decentralization of software development functions.

Some organizations, because of tradition, management style, or external constraints, can never attain true professionalism and shouldn't try. Those, however, that can sustain the needed understanding and commitment will routinely enjoy far higher performance levels in most categories.

## Responsibilities of the Organization

The main responsibility for establishing and maintaining a professional approach rests with the organization. It must:

1. Build, through both recruiting and internal development, a *professional staff* having the ability to perform at the required levels.
2. Maintain for that staff a *professional environment*.

## Building a Professional Staff from Scratch

When we're starting a new organization, we can view staff building from a comfortably theoretical point of view, following a systematic process:

1. Define our exact *charter*, i.e. the products or services we're going to offer.
2. Partition this charter into appropriate hierarchical (or other) *structures* containing precisely defined functions.
3. Derive from these functions precisely defined *roles*. (Some roles will contribute directly to the products and services, while others will provide internal support within our own organization.)
4. Determine the *skills* needed to perform each of these roles.
5. Recruit sufficient *people* with these skills to staff the organization.

This isn't trivial, of course. No matter what methods we adopt, we'll have to make compromises and be satisfied with imperfect results. Nevertheless, we know that the above process will lead to a result reasonably consistent with the organizational objectives and style we're after.

## Professionalizing an Existing Organization

Our task is much harder when we take on a functioning, fully-staffed non-professional organization. We're usually faced with vaguely defined charters and roles, serious skill deficiencies among our staff, and deep-rooted, often unknown human problems. At the same time we must keep providing services without noticeable deterioration.

We should still go through the same first four steps to define a theoretical target, but we can't implement (or even announce) the results until we also have a practical plan for the people we've inherited. Such a plan typically includes:

6. Identifying key roles that we can't fill from the present staff, and recruiting qualified outsiders to fill them. We can announce corresponding organization changes, especially in such staff roles as data-administration or quality-assurance.
7. Reconciling existing salary structures with those needed to attract qualified outsiders. We'll probably have to formalize some of the job descriptions we'll need in our target environment.
8. Diagnosing specific skill deficiencies for the remaining roles and correcting them through a professional development program. These roles include the main body of programmers, analysts, and project leaders. This process, undiplomatically called "upgrading", may extend over many months.
9. Identifying staff members unlikely to perform successfully in our new professional environment, and resolving their career-paths appropriately.

---

<sup>1</sup> Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

The last item is the hardest. To upgrade those with potential is straightforward and satisfying; we'll probably use the same support structures we'll need permanently in our new environment, in particular an internal training or "professional development" program. Dealing with those who can't make it, on the other hand, is a delicate and painful duty that sometimes has no happy solution.

This human problem is sometimes seized upon as an argument either against adopting a professional approach or against some specific methodology component. "Our people were performing satisfactorily before," the argument goes, "but after we changed the environment they no longer could do their jobs. It's not fair to punish them for something we did to them."

Such reasoning misses the point. Rarely do improvements to the environment turn productive people into non-performers. Their performance was *always* poor. What our new professional environment has done is only to make poor performance conspicuous.

An example of this effect in the 1970's was the shift, now widely accepted, in programming from a private activity to a team activity. Many people who had been earning a weekly paycheck for writing and maintaining programs found that they couldn't survive in a world with egoless walkthroughs and rigorously defined module interfaces. In the prior environment they may have taken six weeks to do a three-day programming task and ultimately produced an unmaintainable nightmare, but who ever knew?

The table on the next page summarizes the most important differences in how the two kinds of organizations typically approach staff building.

### **Cultivating Professional Attitudes**

After putting together a staff of people who have either the needed skills or the aptitude to acquire them under our guidance, we still have much to do before our organization can realize the benefits of a full professional approach.

Our staff must *act* like professionals and take pride in being members of a profession. For some, such attitudes are natural. For others, they come as a welcome broadening of horizons. For a few, professional pride remains a foreign notion.

What are the indicators of individual professionalism? What can an employer do to foster them among the staff?

### **Professional Society Membership**

Near the top of the list is participation in a *professional society*. Those who've joined on their own initiative have demonstrated professionalism. Those who haven't done so need our encouragement and they need information. It's appalling to find so many people who've been making their living in data-processing for years but who have never even heard of the ACM or its sister organizations, much less attended a meeting or read a publication.

How can management provide such encouragement?

Above all, we must avoid any hint that the company is reluctant to have its people participate in professional organizations. Some "theory X" managers still express concern that such activities may stir up discontent among otherwise docile personnel.

Second, we must continually make our staff aware of the local chapter activities. We should always post meeting notices on the bulletin boards. When we're planning to attend a meeting ourselves, we can solicit potentially interested people to join us.

Third, the organization should subscribe to and circulate the major publications of the mainstream professional societies.

Fourth, we can support, within reasonable limits, staff members who want to take an active role by presenting papers or by serving as officers. Such support can provide (a) time, (b) word-processing and reproduction services, and (c) travel expenses.

There's one kind of support that we should *not* offer: reimbursing dues or local meeting costs. We want to encourage our people to participate, not bribe them. The true professional views these costs as a routine obligation of his or her professional life, not a fringe benefit. Let's pay our people a fair salary and let them choose freely. Anyone unwilling to invest his or her own money is hardly a real professional.

### **View of One's Role and Skills**

In a professional environment technical staff members view themselves (and should be viewed by management) first as *problem solvers*. The more common view, however, emphasizes their application of specific methods or tools.

In programming, the non-professional orientation most often centers on a specific *language*. "I'm a COBOL programmer," we still hear people say, as if it were some sort of permanent affliction. Misguided recruiting ads perpetuate such stereotypes and convey to the perceptive reader a non-professional image of the organization.

In systems analysis, non-professional people tend to apply standard solution methods to virtually every problem. If the organization follows a standard *life cycle* aimed at developing mainframe application system software, then almost every problem a user brings to that organization will result in establishing a formal project and, if all goes well, the eventual in-house development of application system software that runs on the mainframe computer.

Today most organizations want to consider alternatives to in-house development, like:

1. Packaged application software products.
2. End-user development of part or all of an application system.
3. Experimental or prototype solutions to parts of the user's problem.

In the non-professional environment, such choices are necessarily made very early, often before a project begins. If

you're a user with a problem and you go to the systems and programming people, you'll get a new application system. If you go to the mainframe-oriented "information center", you'll get an entirely different kind of solution. If there's an internal desktop computer support group, you can count on

them to propose a solution based on one or more desktop computers. If you talk first to outside vendors, you get something else again. This is wasteful, confusing, demoralizing, and unprofessional.

<b>KEY DIFFERENCES IN APPROACH TO STAFF BUILDING</b>	
Professional	Non Professional
<b>A. Recruiting</b>	
<b>Selectivity</b>	
Demand top-quality, even if positions remain unfilled Seek 85th percentile or better	Fill given number of available positions. Avoid below-average people
<b>Skills Sought</b>	
Grasp of concepts, theory, problem-solving, broad knowledge Strong oral & written communication	Emphasis on specific languages, machines, other tools Communication skills not stressed.
<b>Experience Level</b>	
Experienced people with demonstrated achievement in various environments Trainees showing evidence of outstanding potential	Batches of trainees with satisfactory academic or work performance Occasional senior people; emphasis on similarity of environment
<b>Reviewing and Evaluation</b>	
Critically examine sample of work. Questioning in depth to elicit free responses High rating for career objective of remaining in professional roles High value on knowledge of the field, membership in professional societies, awareness of trends and current issues.	Rarely look at actual work Questioning emphasizes factual knowledge High rating for ambition to advance in the company out of data-processing Aversion to those with ties to outside organizations or definite opinions on controversial issues. (They may "stir up" the staff.)
<b>B. Training and Professional Development</b>	
<b>Content</b>	
Emphasis on concepts, principles. Stresses ability to think.	Emphasis on factual knowledge of hardware, software, tools, techniques Stresses memorization.
<b>Media</b>	
Reading assignments Group discussions, seminars.	Programmed instruction, multi-media. Lectures
<b>Exercises &amp; Grading</b>	
Largely subjective; no unique "right" answers Instructor evaluates student's mastery	Largely objective; does it work? Instructor certifies student's participation.
<b>Curriculum Planning and Administration</b>	
Tightly integrated with other aspects of methodology Part-time instructors are active practitioners Master plan based on organization's skill needs and individual career-path planning	Little coordination with methodology functions. Full-time instructors are not current practitioners. Individual courses developed or selected out of context
<b>C. Career Path</b>	
<b>Advancement</b>	
Superior performance in technical areas leads toward increased responsibility and scope of influence in technical areas. Salary ceilings recognize productivity ranges and individual's value.	Superior performance as a programmer leads to "promotion" to analyst, and ultimately to supervisory position. Salary ranges based on narrow performance ranges.
<b>Alternatives</b>	
Recognize that some people lack aptitude or interest for technical roles. Welcome new techniques that help expose incompetence; provide alternative career parts for those who fail.	Assume that anyone, with training, can be a programmer Oppose new methods that are "over the heads" of any staff members or that might reveal existing incompetence

Organizations (and people doing research) encourage such parochialism not only by the way they compartmentalize roles and skills, but also by the methods they put forth for measuring performance. It's depressing in the 1980's to see articles seriously proposing essentially the same old quantitative measures we scorned twenty years

ago. We need to develop ways of rewarding our systems analysts for *avoiding* major development efforts. We should reward our programmers for *not* writing yet another redundant routine. We need some way of measuring *problems solved*.

## Professionalizing the Premises

We can infer a lot about an organization's professionalism from its physical work environment. Fortunately, many attributes of the professional office require very little expenditure. Indeed, in some affluent and image-conscious companies we often have to fight to install furnishings that are viewed by some as *negative* status symbols.

Two such low-cost items are bulletin boards and blackboards. Although it's hard for many of us to imagine getting through a day without these basics, a surprising number of companies either prohibit them or don't view them as worth bothering with.

A **bulletin board**, conspicuously placed in a high-traffic area like a coffee room, plays a key daily role in projecting our organization's attitudes. Notices of outside meetings

and courses remind the staff not only of their employer's support but also of their own professional opportunities and obligations. Naturally, someone must be made responsible for keeping the bulletin boards tidy and purged of obsolete notices.

**Blackboards** are an even stronger indicator. Many people do their most creative thinking standing up at a board, even when alone. Some reserve part of their board for semi-permanent status or design information on their current project. Installing a large blackboard in each staff member's office (or "workspace") conveys a clear message that our company understands such habits and that we encourage impromptu problem solving or brainstorming in small groups. Of course, managers' offices must also be so equipped, both to reinforce the impression of organizational professionalism and because managers, too, need blackboards.

KEY DIFFERENCES IN ENVIRONMENT	
Professional	Non Professional
<b>A. Professional Activities</b>	
<b>Membership</b>	
Staff strongly encouraged to participate in ACM, SIM, DPMA, etc. Management sets example by belonging.	Organization indifferent, even unfriendly to professional societies. Management unaware or uninterested.
<b>Participation</b>	
Support for staff members who hold office or serve on committees. Staff encouraged to publish papers, write books, contribute to field.	Hard to justify secretarial services, mail, reproduction, other support for outside activity. Activity mustn't interfere with work.
<b>B. Internal Communication</b>	
<b>Ideas and Opinions</b>	
Staff welcome to contribute ideas and state opinions on relevant issues. Free within the bounds of judgment and discretion; "open" atmosphere.	Those who offer unsolicited views seen as malcontents, troublemakers. Follow chain of command; outgoing communication must be approved in advance.
<b>Controversial Issues</b>	
Constructive debates are encouraged. Shortcomings of current practice are squarely faced and explained.	The staff is shielded from ideas that depart from current policy Shortcomings of current practice are ignored, even censored in documentation.
<b>C. Administration</b>	
<b>Work Assignments and Environment</b>	
Results orientation Flexible working hours Most projects well planned, conducted in an orderly manner; "panic" rare.	Activity orientation Rigid working hours plus overtime Most important projects done in "crash" mode; constant pressure to set aside disciplines so as to "get the job done".
<b>Management Attitudes</b>	
Middle-level managers chosen largely for ability to provide leadership. Instructor evaluates student's mastery	Middle-level managers chosen largely on basis of loyalty and apparent stability. Instructor certifies student's participation.

One of the strongest indicators of an organization's commitment to professionalism is a **publications library**, not just a list of subscriptions but a physical room to which staff members go to consult relevant books, trade journals, professional society publications, and vendors' manuals. Furniture cost is modest: bookcases, a work table, and a few chairs. Support cost, too, is reasonable: the space itself, periodical subscriptions, a budget to buy a few books each month, and a part-time librarian to look after

it. In many organizations this is only a consolidation of costs already being incurred, and setting up a publications library may actually result in a small saving.

For groups spread out in several locations, providing a publications library is more difficult. At the very least, a *central* library should distribute notices of new acquisitions and provide an easy way of circulating material to remote readers.

## Professional Approach to Methodology

An organization's data-processing methodology is its collection of techniques, methods, and tools for defining, designing, building, installing, operating, and maintaining computer-based application systems. The subject matter of methodology is quite broad, including guidance and standards on:

- Project planning and control

- Phased life cycle(s)
- Systems analysis methods
- Data analysis and database design
- Packaged software product acquisition
- Software design techniques
- Program coding and testing
- Operational environment

KEY DIFFERENCES IN METHODOLOGY	
Professional	Non Professional
<b>A. General Approach</b> <b>Basic Philosophy</b>	
Emphasis on quality Standard must serve some obvious purpose.	Emphasis on uniformity. Tendency to standardize everything.
<b>Levels</b>	
Few mandatory standards; only those with global or future impact beyond direct accountability. Majority of material is guidelines, useful information, recommended techniques.	Mostly mandatory standards. Guidelines and recommended techniques must not distract from ability to find out what one "has to do".
<b>Development and Dissemination</b>	
Broad participation of staff in both originating and reviewing content. Highly integrated structure.	Standards committees or standards "czar". Separate components presented out of context.'
<b>B. Presentation</b> <b>Style</b>	
Sharing information with a respected colleague. Mixture depending on nature of material.	Legalistic, military tone; the rules Uniform terse style; e.g. outline format.
<b>Motivation</b>	
Assumes audience wants to learn; eager for help. Emphasis on what one must know to do a superior job.	Assumes audience resistant. Emphasis on satisfying enforcement ("QA") and audit procedures.
Explains the purpose of any standard.	Avoids excess "verbiage"; just gives the bare standards.
<b>C. Enforcement</b> <b>Quality Assurance</b>	
Clear separation between a quality review (Is there a better way?) and enforcement (Did we follow the rules?) Friendly, helpful spirit.	Confusion between quality review and standards enforcement. Tendency toward second guessing on matters of individual choice or style. Adversary atmosphere.
<b>Deviation Approval</b>	
Emphasis on what's best for the company. Resolution based on comparative evaluation of costs, benefits, risks. Everyone understands rationale behind final resolution.	Emphasis on conflicting goals of the organizational units. Resolution emphasizes organizational power and influence. Management decisions convey images of winners and losers.

Recent trends toward more powerful tools and more intimate user interaction during development have generated some backlash against methodology in general. Often such reactions are justifiably intensified by misguided methodology extremism illustrated by:

- excessively formal and complex procedures, e.g. in a large life cycle.
- arbitrary standards that contribute little if anything to performance
- overzealous, often adversarial quality assurance reviews.

Despite these backlashes, *good* methodology remains indispensable to today's software development organization.

The *style* of an organization's methodology reflects its orientation to a professional approach. In a non-professional (sometimes called "military") methodology just about everything is a *mandatory standard* or a *policy*. Standards manuals keep warning the staff either *always* to do something or *never* to do it.

On the other hand, under the professional approach most components are conventions, guidelines, or recommended techniques, not mandatory standards. We prefer flexible methodology because we *trust* our staff to apply good

judgment, consistent with their levels of experience and responsibility. Experience confirms that such flexibility in the hands of competent staff leads to superior performance in terms of quality, manageability, and productivity.

### **Staff Functions and Overhead**

The professional approach can't compete in two popular measures of organizational performance:

1. Hourly rates for professional services.
2. Ratio of billable to total hours.

Because of the greater number of internal support functions a highly professional organization must allocate more overhead for such things as methodology administration, quality assurance, training, publications library, program module library, data administration, and internal consulting. Hourly rates for professional services will include these costs and will therefore be significantly higher than in a non-professional "body shop" environment. These rates should be easy to justify in terms of the value delivered, but the lack of solid quantitative measurement may make it hard to persuade some doubters.

### **Austerity Threats**

These same overhead functions are especially vulnerable to budget cuts in bad economic times. Upper management may not understand the connection between these functions and the organization's performance, or they may think that the organization can coast for a year or two with sharply reduced staff support

In fact just the opposite is true. These "overhead" functions, assuming they're being performed competently, should be the *last* items trimmed from a tight budget because of the high multiplier on their effects. An individual programmer writing superb code furiously all day can at most affect only the project he or she is working on. The impact of a well-chosen piece of methodology, on the other hand, may be felt in *every* project.

Sadly, experience shows that the momentum of professionalism is very hard to restart once it has been cut off. The world of large-scale data-processing is littered with the ruins of organizations that once made a proud commitment to professionalism and flourished for several years, only to lose it all in a reorganization, a merger, or a budget squeeze.

Fortunately, there are a few ways of making the support functions more conspicuous and clarifying the connection between them and the products and services of the organization. Rigorous role definitions and strict measures of performance, for example, are a good defense against heavy-handed cuts by any but the most irrational manager.

## **Categorizing organizations**

In these discussions we've drawn a sharp contrast between two radically different philosophies. In reality, of course, no organization is 100% one thing or the other. When we speak of a "professionally oriented" organization we mean one that exhibits *most* of the characteristics associated with the professional approach.

While organizations are not 100% professional or 100% non-professional, neither can they stay right in the middle. The great majority of data-processing organizations strongly emphasize one approach or the other.

### **When Non-Professional is Appropriate**

Obviously the professional approach is *better* in almost every way, but it's not necessarily better for every kind of organization. Some companies or institutions will be better off settling for predictable non-professionalism. Among the practical factors pushing an organization in that direction are:

- Rigid policies imposed from an outside source like a parent company or a regulatory agency. An example would be below average salary ranges.
- A deeply ingrained management style emphasizing snap judgments and intuitive rather than rational decisions.
- Expected continuing decline in business volume or a lack of new applications.
- High level of satisfaction with the status quo among users and ultraconservative upper management.
- A large tenured staff lacking upgrade potential.