Do you have what it takes to be a Software Professional?

Quick! Think of three oxymorons! How about military intelligence (I was in the military), thundering silence (I'm not very silent), and computer professional (I wish it wasn't an oxymoron).

Why do you suppose someone would like to be a "professional"? Some people think that as a professional, they will receive some form of reward or recognition. Some people think being a professional means the professional is better than average. Professionalism often involves extra effort. Once we achieve the "professional" status, we are rewarded (usually more money or some suitable substitute) and we get recognition from our peers, clients, and the industry in general.

The flip side of the coin is "Why do we want to do business with 'professionals'"? Dealing with professionals gives us that added assurance that things will be done correctly. Our comfort level is "upped". If you don't believe me, look at the alphabet soup of certifications desired in the computer want ads in this Sunday's paper.

So What's A Professional?

As I think about being a computer professional, I get hung-up on the professional part. I'm assuming that we share a common enough definition for a computer (yes, I know this can be dangerous), but how do you define professional? Common definitions would include [1]:

1. A person who is engaged in a profession. Some generally recognized professsions[2] are medicine, law, law enforcement, teaching, civil and professional engineers, and the military. The characteristics of these types of professions are:

   - Society has given the members of the profession the exclusive right to engage in the profession. Non-members are barred from practicing the profession.
   - Professions require specialized education and training (which is determined by the profession)
   - Professions are self-policing. They have a code of ethics and behavior and the power and will to enforce that code.

   While I wish it were, it is obvious that creating software is not this type of profession.

2. A person who does their job with great skill.

   The convenient portion of this definition is that it allows us to define professional within the context of a specific instance. The problem is how do we convince those we've not worked with that we're "professional"? I think this definition holds some hope for us. An Alta Vista search on software AND professionalism returned a mere 56,970 hits. There definitely seems to be some smoke, but can we find a fire?

3. A person who engages in an art or sport etc., for money, especially as a means of livelihood.

> Here I believe we have the lowest common denominator for “professional”. This doesn’t require a profession (which we don’t have) or great skill (and some of us seem to be lacking this also). The good news is that we get paid. The bad news is of course, that we still get paid regardless of how well we fit definitions 1 & 2.
>
> Cynically, I'd say that this is where most "computer professionals" are functioning. It's easier to excuse our lapses in definitions, our lack of care to others, and our general attitude of wanting to show that we're "hot stuff". This is where you'd place the consultant who doesn't deliver on time, the employee who only writes the code he's told to, and the firms which fight over clients and turf like raw steak in a dog food bowl.

So what to do? Definition 1 is out and definition 3 is too inclusive. What can we do with definition 2?

Professionalism Models

In Tom DeMarco’s forward to “The Responsible Software Engineer”, three models of professionalism are described. The "Model Zero," the "3-P Model" and the "4-P Model." I would like to add a fourth; at a lower level than Mr. DeMarco's lowest level. My contribution to the list is probably the most bothersome. I call it the "Clueless Category." Can you identify yourself and other computer professionals in the following models?

Clueless Category: The members of the Clueless Category are, well, clueless. They don’t have a clue that there are such things as standards, conventions and best practices. This means these people do what seems best at the time. The decisions are single point considerations without much thought. I’ve been around long enough to see that a number of computer “professionals” fall into the Clueless Category.

***Model Zero*:** Tom DeMarco says: “I intend a pejorative sense to this name, since the attitude represented by Model Zero is retrograde and offensive . . . but nonetheless common. In this model, the word ‘professionalism’ is a simple surrogate for compliant uniformity.” [3]

This loosely translates to you doing what your boss told you to do since he’s doing what his boss told him to do since he’s doing … and so on. Should you make the poor choice to engage in non-conformity, you’ll certainly be deemed “unprofessional”. Early in my career, my management felt that (male) programmers should wear shirts and ties. Wearing a shirt and tie did nothing for my professional development.

3-P Model: The 3-P Model uses three characteristics to describe professionalism. "The three characteristics are:

- Proficient: Whatever it is that a professional does, he/she must do it with deftness and agility, and with the skill born of long practice.
- Permanent: The long practice comes from the permanence of the professional’s calling…

- Professing: Finally there must be some act of involvement by which the professional declares his/her intention to be, now and forever, a part of one chosen calling. The act may be a public ceremony of it may be a simple, private resolution of the form:

  <name of profession> EQUALS me." [3]

Tom DeMarco goes on to say: "The 3-P Model still lacks something, an ethical dimension. People we think of professionals are governed by some kind of code. They know their profession gives them opportunities for wrongdoing, and they know what they will and will not do for ethical reasons. …a fourth P-term:

- Promise-keeping: Professionals make certain promises to themselves (sometimes to the public at large) about what they will and won't do. Professionals keep those promises". [3]

Years ago I was told that to be successful all I had to do was find out what I liked to do, and then find someone to pay me to do it. Even after 20 years, I still can't think of anything else I'd rather do than software development. I like the 3-P Model since it calls from a higher plane than "show me the money". But 3-P still won't get us to definition 1. For that we need Tom's fourth P.

If you happened to need a handy set of promises, you might want to try looking at the Draft Software Engineering Code of Ethics.[4] You may also want to take a look the code of ethics of the Independent Computer Consultants Association. Where there isn't a total match, you'll find that they're after the same thing: behavior, which is responsible to society, our clients, our employers and our associates.

Parts to a Profession

A code of ethics would be a start. We can be professional, and work toward creating a profession. What would it take to create a profession? There are three fundamental issues we must deal with, based on the three prongs of the definition. This is where the members of our profession differ

1. Body of knowledge: Specialized training requires us to specify what it takes to be a computer professional. For every person, not on a case-by-case basis. Some feel this is a problem for those who feel we've learned some unique way of delivering especially good software. We don't agree on what the body of knowledge is. We don't even share computer languages, operating systems or basic methodologies.

   The body of knowledge required to be a computer professional isn't defined in computer languages, operating systems or basic methods. The body of knowledge is the foundation on which the languages, systems, and methods exist. Think of the *"what we need to know to do the job"*, not the *"which particular tool do we use to do it"*. Consider the medical profession. Obviously a brain surgeon and a podiatrist have very different specialized training. Yet, they both belong to the same professional organization. It seems to me then a network administrator and a database administrator (who granted have different specialized training) should be able to part of the same professional organization.

2. Certification: Who will we endow with the power to decide who is a professional? Any good ideas for this one are obliterated by the fact that every current attempt at defining testing for the software profession is only supported by a small group, which mostly does not include employees of major companies. So what is the "required core body of knowledge"? Perhaps since I don't program very much, I can't even fulfill such a requirement. Maybe I can manage it, but not do it. Or maybe everyone needs to be able to explain Codd and Date's principles of databases, whether they work with them or not.

   This is an difficult topic. Once again the critical point is the “required core body of knowledge”. It is going to take some effort to define the body of knowledge. Once the body of knowledge is defined, then a way of certification can be determined. Consider the Project Management Institute (yes, I know it’s not a ‘profession’ … yet). They have managed to distill and publish (available for free download from their website) a body of knowledge necessary to properly manage any project. The “certification” involves a test, and documenting when and where you’ve had experience in the field.

3. Policing. Everyone seems to have his or her own definition of right and wrong. In a world of fairly stable values, the medical, legal and accounting professions could substantiate basic requirements. But even those are carefully matched to legal protections. The existing professions primarily seem to members based on felony convictions.

   As computers (in every sense of the word) become even more prevalent and are used in more critical applications, society will demand that those involved in the computing field be policed. The arguments of “different definitions of right and wrong” and “nobody else does a real good job of policing” are diversionary tactics. Policing will happen. We have a choice: will we lead the discussion and efforts, or will a disaster and the resulting government legislation/mandates lead us?

As computer professionals, we should lead the effort to describe a Body of Knowledge, develop a certification procedure, and define a body that administers certification.

When we have done these things, then "computer professional" will no longer be an oxymoron.

I would like to thank Sharon Marsh Roberts and Johanna Rothman for their comments.

[1] definitions for professional (noun) from Webster’s New World Dictionary, Third College Edition. - © 1991

[2] The discussion on professions is one that Rich Cohen routinely drills in the DCI CASE Forum on CompuServe.

[3] Tom DeMarco’s professional models are discussed in “PROFESIONAL AWARENESS IN SOFTWARE ENGINEERING”. I found the article at: http://www.atlsysguild.com/Site/Tom/Professionalism.html

[4] **Draft Software Engineering Code of Ethics by the IEEE Computer Society and ACM**. On line at: http://www.computer.org/tab/seprof/code.htm

PRINCIPLES v 2.1 (I've only included the preambles for each Principle Section)

Principle 1: PRODUCT

Software engineers shall, insofar as possible, assure that the software on which they work is useful and of acceptable quality to the public, the employer, the client, and the user, completed on time and at reasonable cost, and free of error.

Principle 2: PUBLIC

Software engineers shall, in their professional role, act only in ways consistent with the public safety, health and welfare.

Principle 3: JUDGMENT

Software engineers shall, insofar as possible and consistent with Principle 2, protect both the independence of their professional judgment and their reputation for such judgment.

Principle 4: CLIENT AND EMPLOYER

Software engineers shall, consistent with the public health, safety, and welfare, always act in professional matters as faithful agents and trustees of their client or employer.

Principle 5: MANAGEMENT

A software engineer in a management or leadership capacity Shall act fairly and shall enable and encourage those who they lead to meet their own and collective obligations, including those under this code.

Principle 6: PROFESSION

Software engineers shall, in all professional matters, advance both the integrity and reputation of their profession as is consistent with public health, safety, and welfare.

Principle 7: COLLEAGUES

Software engineers shall treat all those with whom they work fairly and take positive steps to support these collegial activities.

Principle 8: SELF

Software engineers shall, throughout their career, strive to enhance their own ability to practice their profession as it should be practiced.

[5] **ICCA Complete Code of Ethics** on line at http://www.icca.org/ethics.htm

- Consultants will ensure that to the best of their knowledge they can complete the project in a professional manner both in terms of skills and time.

- Consultants who are unable to professionally complete part or all of the contract, will be forthright and will offer to aid the client in finding resources to complete the contract satisfactorily.

- Consultants will be honest and not knowingly misrepresent facts.

- Consultants will not engage in contracts that are in violation of the law or that might reasonably be used by the client to violate the law.

- ICCA member firms, their principals and employees will uphold the principles of the ICCA and not commit acts discreditable to the ICCA.

- Consultants will divulge any potential conflicts of interest prior to accepting the contract or as soon as possible after the conflict is discovered.

- Consultants will only represent opinions as independent if they are free from subordinated judgment and there is no undisclosed interest in the outcome of the client's decision.

- Consultants will not take advantage of proprietary information obtained from the client.

- Consultants will safeguard any confidential information or documents entrusted to them and not divulge any confidential information without the consent of the client.

- Consultants will keep the client informed of any matters relating to the contract even if the information is unfavorable, or may jeopardize the contract.

- Consultants will install and use only properly licensed software on their systems as well as the client systems.

- Consulting firms who use subcontractors can use a non-compete clause to restrict the subcontractors from working directly with their clients for a specified period of time. If the prime contractor uses a non-compete clause in their contracts, the term of the non-compete should be one year or less.

- Consulting firms will not compel independent computer consultants to work as employees when they prefer to work as independent contractors.

- Consultants will devote a significant portion of time in continuing education

EPILOUGE:

Here it is four years later. I still don’t feel we have a profession, but I do believe there is definite progress. Perhaps most noteworthy, is that the URLS for the IEEE and ICCA code of ethics are still VALID! In Internet time, this means they’ve been there forever. The IEEE SOFTWARE ENGINEERING CODE OF ETHICS AND PROFESSIONAL PRACTICE is now version 5.2 not 2.1. It is heartening to see the significance placed on professionalism.

Additionally, in the software realm, a software engineering body of knowledge has been developed. Check out http://www.swebok.org While this doesn’t embrace the breadth of “computer professional”, it is a start.

In terms of certification, the IEEE Computer Society has developed a Certified Software Development Professional designation. Check out http://computer.org/certification/ This addresses the underlying bedrock of software development, not which particular computer language or OS is being used.