Status of the task force for defining the required body of knowledge and recommended practices

Patricia Douglas (pjd@vnet.ibm.com)

As Capers Jones outlined in the article "Legal status of software engineering" [*Computer*, Volume 28, No. 5, May 1995, pages 98-99], one of the factors associated with recognized formal professions is "a well-defined body of knowledge, and often many subsets of more specialized knowledge with an academic curricula to transfer that body of knowledge to students. The current work being done within the IEEE Computer Society and the ACM task forces is moving towards a goal of making software engineering the 37th engineering profession.

There are three current task forces attached to the joint steering committee for the establishment of software engineering as a profession. The task forces were chartered by the steering committee to accomplish the following recommendations: 1. Adopt standard definitions 2. Define required body of and recommended practices (in electrical engineering, for example, electromagnetic theory is part of the body of knowledge while the National Electrical Safety Code is a recommended practice.) 3. Define ethical standards 4. Define educational curricula The steering committee has also begun to study certification or licensing in other engineering professions.

The task force for establishing the software engineering body of knowledge and recommended practices has been moving forward with the author as chair and recently a co-chair joined me. Anthony Cocchi brings his wealth of experience in systems architect, software design and programming to this work. Tony's current job responsibility is as a senior designer/programmer from IBM Research.

The approach being used by the task force is a series of surveys. The surveys will be used to define piece by piece the body of knowledge and practices at different levels of knowledge/skill needed at different points in the career of a Software Engineer. These surveys will capture the categories of components of knowledge. These components of knowledge fall into two categories: generic knowledge, such as mathematics, science engineering science, etc.; and knowledge that is specific to Software Engineering, such as software analysis, software architectures, etc. In each area, we will address static knowledge as well as dynamic knowledge (thins we do).

Work began over a year ago using focus groups to establish an exhaustive list of all possible areas that should be considered in software engineering. Continued work was done to organize this input to establish categories that should be included in an international survey. The survey methodology was chosen because it allowed for the greatest coverage throughout the industry. The format of the survey will allow us to analyze the common core of what is currently going on throughout the world as well as those areas that are particular in an industry segment.

Creating the task statements within these areas is being accomplished by organizing small groups of experts and creating the task statements that accompany each area. The focal point of this work is the establishing of industry sub-chairs who will have the following

responsibilities:

- Serve as focal point in your industry sector
- Facilitate groups in your area to write task items in identified software areas
    - Arrange for the right people to be at the task writing workshop
    - Arrange for the location/ send out notices/ serve as host for the task writing workshop
    - Identify an individual who can observe the process and conduct further workshops for writing task statements as necessary
    - Serve as project manager within your industry sector to see that task items are completed on schedule and returned to chair
- Identify strategic individuals to assist in distributing the survey and collecting the completed survey
- Work with strategic individuals to assure proper coverage for distribution of survey
- Serve as a communications point for work in progress
- Identify mechanism within industry sector to publicize ongoing work and gain support

We have touched on only 58 topics at this time which is only about 10% of what we need. We need people within the industry to serve as sub-chairs for this work. Our goal is to have the input for the creation of the survey completed by 3Q 95. If you can participate in this work, please contact Patricia Douglas (pjd@vnet.ibm.com) or Tony Cocchi (tony@watson.ibm.com).

SIDEBAR — Sample Survey Question

This is an example survey question, formatted roughly as it might be for the actual survey instrument. The example illustrates the kinds of questions that we need to create.

For each statement, check the box for each category of software professional for whom the statement is true in your organization.

Object-Oriented Programming

A software professional in my organization should be able to:	Novice SE (1-3 years)	Experienced SE (7-10 years)	Software Project Manager	Specialist (Scientist Domain Engineer)	Not applicable in my organization
1. Define "object", "class", "inheritance", and "polymorphism".	❒	❒	❒	❒	❒
2. Read and understand programs written in an object-oriented language.	❒	❒	❒	❒	❒
3. Write simple object-oriented programs.	❒	❒	❒	❒	❒
4. Maintain object-oriented programs.	❒	❒	❒	❒	❒
5. Select appropriate classes from a class library, and define appropriate subclasses for a given application.	❒	❒	❒	❒	❒
6. Create an application framework and class library for a particular problem domain.	❒	❒	❒	❒	❒
7. Evaluate object-oriented programs and identify opportunities for improvements.	❒	❒	❒	❒	❒

The table reports percent of survey respondents (n=68) who indicated that "a software professional in my organization should be able to" do each of the seven tasks listed, for each of four job categories: Novice SE (1-3 years), Experience SE (7-10 years), Software Project Manager, and Specialist (e.g., Scientist, Domain Engineer).

Displaying the percent of repondents who respond "Yes" (or "True") to each question either in a

table or in graphical form is one parsimonious way to view results from this type of questionnaire.

Object-Oriented Programming Percent Responding "Yes" to each of Seven Questions for Five Categories	Novice SE (1-3 years)	Experienced SE (7-10 years)	Software Project Manager	Specialist (Scientist Domain Engineer)	Not applicable in my organization
1. Define "object", "class", "inheritance", and "polymorphism".	79.4	86.8	55.9	70.6	2.9
2. Read and understand programs written in an object-oriented language.	73.5	86.8	32.4	57.4	5.9
3. Write simple object-oriented programs.	80.9	85.3	11.8	55.9	5.9
4. Maintain object-oriented programs.	58.8	83.8	8.8	41.2	5.9
5. Select appropriate classes from a class library, and define appropriate subclasses for a given application.	27.9	83.8	8.8	57.4	7.4
6. Create an application framework and class library for a particular problem domain.	8.8	70.6	10.3	67.6	5.9
7. Evaluate object-oriented programs and identify opportunities for improvements.	5.9	76.5	13.2	57.4	7.4

In the Object-Oriented Programming area, it is clear that the novice SE is generally expected to be able to the statements representing the lower levels of Bloom's taxonomy, but increasing less at the higher taxonomic levels. This is true but much less markedly so for experienced SE's. For the software project manager, little is generally required in Object-Oriented Programming. Certain interesting features of the data can be pointed out. For example, the Specialist shows a dip in percent for question 4 "Maintain object-oriented programs" -- evidently program maintenance is more likely to be left to experienced SE's.

Object-Oriented Programming

Percent Responding "Yes" to each of Seven Questions for Five Categories