Software Engineering Education (SEEd)
Peter B. Henderson
Department of Computer Science & Software Engineering
Butler University
Indianapolis, Indiana 46208 USA
phenders@butler.edu

To date I have not received any comments regarding Don Reifer’s article “Educating Software Engineers: An Industry Viewpoint” in the last SEEd column. Perhaps there are a few letters to the editor. By the way, I would like to complement Will Tracz SEN editor. He does an excellent job pulling each issue of SEN together, and we are indebted to the time and effort he contributes to making SEN very readable and informative.

The theme of this column follows that of the last “Educating Software Engineers.” As software engineers we have individual and collective views regarding how software engineers should be educated and could make a fairly comprehensive list of bullet points of the goals of software engineering education.

Recently while searching the web I ran across the following list of goals written by David Parnas and his wife Lillian Chik-Parnas, who is an educator. They had developed this for a company in Warsaw, Poland that wanted to start a new private University level education, and I am reprinting it here with their permission. I hope you find it as useful and informative as I have.

Goals for Software Engineering Student Education
David Parnas and Lillian Chik-Parnas

1 General

1.1 Ethical behavior and social responsibility
1. Graduates who have been taught to be habitually honest about what they have accomplished.
2. Graduates who are quick to give others credit for their contributions.
3. Graduates who understand their responsibilities to their employers and customers.
4. Graduates who understand their obligations to society at large.

1.2 Personal insight and maturity
1. Graduates who are able to perceive commonalities between things that appear quite different.
2. Graduates who are able to understand their own limitations and capabilities and work within them.
3. Graduates who are able to handle stress and know to get help or relief when they need it.
4. Graduates who understand the conditions that they need to work best, and can achieve a personal balance.
5. Graduates who are able to recognize, understand, and respond properly to (e.g., recover from, their own failures).
6. Graduates who are able to handle criticism without becoming defensive.

1.3 Professional insight and maturity
1. Graduates who know what it is that they do not know.
2. Graduates who are not afraid to explore and inquire about topics outside their area of expertise.
3. Graduates who know how to manage their time.
4. Graduates who understand the way that long-term projects progress and are able to make measurable progress without taking unwise short-cuts.
5. Graduates who understand when to take risks and what risks are worth taking.
6. Graduates who habitually attempt to anticipate problems and prepare for them.
7. Graduates who understand the need for trade-offs and know how to make such decisions.

1.4 Flexibility
1. Graduates who are flexible, i.e. able to change when their environment changes.
2. Graduates who will do something that they were not told to do when that something is the “right” thing to do (i.e., graduates who will show constructive initiative when needed).
3. Graduates who are able to analyze each situation and react by finding or defining an appropriate method to solve the problems. This requires that they understand the conditions under which a method is applicable and when one must choose or invent a different method.
4. Graduates who are able to transfer what they have learned in one area to other, possibly very different, areas?

1.5 Thinking, learning and working skills
1. Graduates who habitually work to extend their own capabilities (life-long self-directed learners).
2. Graduates who are able to work well within groups.
3. Graduates who are able to form and lead groups.
4. Graduates who habitually investigate the history and detailed definitions of ideas and concepts which are new to them.
5. Graduates who habitually demand precise definitions of terms.
6. Graduates who understand when and how “trial and error” is an appropriate way to explore a question.

7. Graduates who are able to learn to habitually apply principles that they have first learned or formulated consciously.

8. Graduates who habitually apply the principle divide and conquer to complex problems.

9. Graduates who are able to question conjectures and confirm or refute them.

**2 Communication**

1. Graduates who are able to organize talks, documents, papers so that they present information in a way that is appropriate for the intended audience.

2. Graduates who have a command of English sufficient to communicate about their work.

3. Graduates who are prepared to learn new natural languages when it would be useful.

4. Graduates who are skilled in communication in a group, especially about goals, assignments and progress.

5. Graduates who know how to present complex topics in small, easily understood sections.

6. Graduates who are able to express consciously the principles that they unconsciously apply.

**3 Mathematics**

**3.1 General Mathematical Skills**

1. Graduates who know how to think carefully and logically.

2. Graduates who know how to abstract, finding more general, reusable concepts.

3. Graduates who are familiar with many useful concepts that have already been studied in mathematics.

4. Graduates who know how to make models of reality by abstracting from irrelevant details.

5. Graduates who know how to apply mathematics and other abstract models in their work and daily life.

**3.2 Applying Mathematics in Software Development**

1. Graduates who are able to use mathematical logic to describe software states and functions.

2. Graduates who understand basic logical concepts, are able to generate proofs, and know how to choose and use logic-based tools such as automated theorem provers.

3. Graduates who understand how to apply discrete mathematical concepts such as graphs, trees, relations, when developing software.

4. Graduates who know how to apply numerical mathematics.

5. Graduates who know how to apply symbolic mathematics.

**4 Software Development**

**4.1 Software Fundamentals**

1. Graduates who understand the fundamental mathematical laws governing what we can do with software, and the physical laws governing the application and design activity.

2. Graduates who know what makes software development difficult.

3. Graduates who understand how software is developed now, both right and wrong.

4. Graduates who understand why the “man-month” is not a unit of software task complexity.

5. Graduates who know how to reduce the amount of effort it takes to create and maintain software.

**4.2 Software Technology**

1. Graduates who know how to use current tools and how to learn to use new tools.

2. Graduates who are masters of the craft of programming.

3. Graduates who know who to produced networked applications.

4. Graduates who know how to choose the appropriate program development tools.

**4.3 Software Design and Analysis**

1. Graduates who are able to write programs that satisfy specifications that have been provided to them.

2. Graduates who are able to prepare program tests given specifications.

3. Graduates who are able to inspect programs to determine whether or not they meet specifications.

4. Graduates who are able to “modularize” software so that the modules can be developed, tested, and understood separately.

5. Graduates who are able to design software products and software components by writing interface specifications.

6. Graduates who know how to build configuration-and device-independent (distributed) software.

7. Graduates who know how to build software that must deal with concurrent activities.

8. Graduates who know how to build software that meets real-time deadlines.

9. Graduates who know how to estimate/limit the accuracy of numerical results in scientific (e.g. physical) calculations.

10. Graduates who know how to choose and design effective algorithms.

11. Graduates who know how to solve a problem before the prob-
lem is specified and will know how to determine a complete and consistent set of requirements.

12. Graduates who understand how to organize data for products that must store and process large amounts of data.

5 Scientific Method
1. Graduates who know what it means to know something and what methods allow us to know something.
2. Graduates who understand how to design and conduct experiments.
3. Graduates who understand how to draw correct conclusions from observations.

6 Management, project planning, and economics
1. Graduates who understand basic legal and business.
2. Graduates who understand basic accounting, tax issues, etc. enough to operate small company or to work with experts on specialized software.
3. Graduates should know how to apply methods of protecting information against loss and disclosure.
4. Graduates who understand information protection policy issues well enough to ask the right questions of the policy makers.
5. Graduates who understand intellectual property rules enough to know their rights and obligations.
6. Graduates who know how to plan projects, defining milestones and setting deadlines for meeting those milestones.
7. Graduates who are familiar with various approaches to software-project cost estimation.