

# THE SOFT SIDE OF SOFTWARE ENGINEERING

## *NIFTY COURSE ASSIGNMENT PRESENTATION*

Chuck Pheatt  
Department of Computer Science  
Emporia State University  
Emporia, KS 66801  
620 341-5637  
*cpheatt@emporia.edu*

### **ABSTRACT**

Software Engineering (SE) today is much more a challenge of organizational and methodological improvement than a technological one. Most software engineering courses and texts provide an excellent framework for young engineers to understand the nature of the software development process and weigh alternative approaches in practicing their craft. However, most standard texts on the subject pay little or no attention to the human factors that are operating within technological organizations. This talk will present a number of case studies based on the book *Managing Humans*. These case studies can be used as an adjunct to conventional SE course content to address the lack of “human centered” content.

### **INTRODUCTION**

Software engineering (SE) is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software [1]. The study of software engineering includes the knowledge, tools, and methods for defining software requirements, and performing software design, construction, testing, and software maintenance tasks. Most SE instructors would agree with these definitions, but fail to recognize that these lofty practices are implemented by highly trained and talented programmers who embody all the emotions, conflicts and imperfections of humans.

The software engineering community has begun to recognize that software development is a collaborative activity, and much recent research has begun to appear in this domain [2]. Studying and supporting the social side of software engineering is not a new idea [3]. Software engineering was one of the original contexts where communication and coordination breakdowns were identified as one of the major causes of problems in software development.

Missing from the aforementioned discussions are directly applicable strategies that new software engineers can use in the workplace. In his 2007 book [4], Michael Lopp, a veteran engineering manager from Silicon Valley, explores the difficulties of managers and the managed. Lopp has worked at a variety of companies including Apple Computer, Netscape Communications, Symantec Corporation, Borland International, and brings his experience set to the discussion.

In the 209 page book, Lopp tells us what the manager should be doing for his employees. He also tells what the employees should be doing to get the most benefit from

their manager. He discusses the phenomena of the Monday morning "freakout", and what's really being said behind the emotional outburst. He categorizes individuals as *Incrementalists* and *Completionists* and discusses the ramifications of dealing with these individuals in a team environment.

This talk will present a number of case studies based on Lopp's book. These case studies can be used as an adjunct to conventional SE course content to address the lack of "human centered" content in most SE courses.

## **REFERENCES**

- [1] IEEE Standard Glossary of Software Engineering Terminology," IEEE std 610.12-1990, 1990.
- [2] Glass, R.L., *Facts and Fallacies of Software Engineering (Agile Software Development)*. Addison-Wesley, 2002.
- [3] Grudin, J. Computer-Supported Cooperative Work: History and Focus. *IEEE Computer*, vol. 27, no. 5, May 1994, 19-26.
- [4] Lopp, M., *Managing Humans: Biting and Humorous Tales of a Software Engineering Manager*, APress, 2007.