Technical Communication in Your Field

STEVEN PAUSTIAN, New Mexico Institute of Mining & Technology

TODO [the absract goes here, dummy]

Categories and Subject Descriptors: C.2.2 [**Technical Communication**]: Computer Science

General Terms: Comments, Communication, Issue Tracking, Technical Language, Version Control

Additional Key Words and Phrases: Technical Communication, ACM Small Format

ACM Reference Format:
Steven Paustian 2015. Technical Communication in Your Field *ACM Ex. Journ.* 0, 0, Article 00 (March 2015), 4 pages.
DOI: http://dx.doi.org/00.000/0000000.0000000

1. INTRODUCTION

Computer Science (CS) is the study of computation and its applications. CS majors are employed in a wide variety of positions post-graduation; prominent job titles include Application Developer, Information Technology Architect, and Systems Administrator. [Payscale.com 2015]

The CS field contains a broad array of delivery mechanisms for technical information. Many are comparable to other fields, including peer reviewed journals, memos, and technical reports. In addition, it involves a number of communication tools that are unique such as issue tracking and code commenting.

To further explore technical communication in the field, the author interviewed a recent graduate and a seasoned professional programmer. A discussion of these interviews can be found in [2]. Three examples technical documents were gathered and will be discussed in [3]. Two secondary sources regarding technical communication in the field were also found and will be discussed in [4].

2. INTERVIEW ANALYSIS

2.1. Professional Programmer

Frank Germano, a professional programmer of 30 years, was a wealth of information regarding technical communication. As a former vice president of a software engineering company, and current lead developer, he noted technical communication requirements may vary widely through one's career.

As a lowly programmer, most technical communication involves code commenting, software documentation, and issue tracking (verbose descriptions of buggy code). Lead developers and project managers serve as a bridge between programmers and non-technical management, and their communications reflect this. They tend to handle

This work was intended, designed, and written for ENGL341-02, Spring 2015.
Author's address: S. Paustian, Computer Science Department, New Mexico Institute of Mining & Technology, Socorro, NM.

DOI: http://dx.doi.org/00.0000/0000000.0000000

state diagrams, flowcharts/UML diagrams, memos, and (sometimes) contracts. He revealed that larger firms will typically include a Technical Writer in teams working on large project. This person's sole responsibility is creating, editing and interpreting technical documention for the software the team is currently writing.

The bulk of a programmer's non-coding time is spent on internal communication, namely emails.

TODO[Add time limitations, revision process]

2.2. Recent Graduate

Jon Zingale is a recent graduate from The University of Texas as a math major with a CS minor. He was hired by a security firm in Silicon Valley. He reports that a large portion of his time is spent on communication; mostly emails, memos, and documenting security problems both in code and an internal issue registration system. He rarely needs to convey technical information to non-technical personel, though he was recently asked to give a presentation to client regarding security holes discovered by his team.

TODO[flesh this out]

3. EXAMPLE DOCUMENTS

Three example documents were obtained from Frank and Jon. They are: an example of properly commented code, an example of issue documention, and a monthly memo sent to clients.

TODO[add margins to table, fix table placement, finish table cells]

Document Type	Audience	Purpose	Style	Formating
Issue Tracking	Management tracking progress on a bug fix. Internal programmers assigned to fix the issue.	Provide specific technical and tracking information regarding a bug that needs to be fixed.	3	Conforms to internal specifications designed for clear communication of technical information.
Customer Memo	Current customers and the general public. Company employees.	Provide customers and employees with "state of the union" information about the company.	3	Colorful and easy to read. Different font weights and colors highlight important information and what it relates too. Information is regarding new customers and new software functionality.
Code Comments	Programmers and other employees working directly with the code.	Explicitly explain code parameters, usage, and how the code works.	3	Conforms to internal commenting standards.

4. SECONDARY DOCUMENTS

TODO[Consider adding brief intro]

4.1. Comments Are More Important Than Code

Communication inside code is unique to programming. Programmers often talk of making code "self-commenting," but this is nearly impossible to do. Code is, by definition, complicated and difficult to understand. Comments are an essential tool for communicating an algorithm's intent, usage, and implementation. In other words, good comments explain how and why.

Raskin argues that code comments are not only important, but essential for code to be reliable and maintainable.[Raskin 2005] The problem is, enforcing strict commenting guidelines is cumbersome and limits the amount of code a programmer can write

in a given unit of time. Raskin argues that despite this, the benefits of rigorous comments far outweight the costs. Rebuilding code with good documentation is far easier than without. In fact, writing any code is far simpler if documention exists beforehand. [Raskin 2005]

4.2. Agile Development

Agile software development is a software development that stresses working code over ideas, and effective person to person communication over documents and long response times.[Highsmith and Cockburn 2001] This section will focus on the communication aspects of Agile development, and how those concepts assist in creating effective and robust software.
TODO[flesh this out]

5. CONCLUSION

TODO[fill this out]

ACKNOWLEDGMENTS

The author would like to thank Frank Germano and Jon Zingale for their interviews.

REFERENCES

Jim Highsmith and Alistair Cockburn. 2001. Agile software development: The business of innovation. *Computer* 34, 9 (2001), 120–127.

Payscale.com. 2015. Popular Jobs for Computer Science Majors - College Salary Report. (2015). http://www.payscale.com/college-salary-report-2014/choosing-a-major/popular-jobs-by-major/computer-science-and-math

Jef Raskin. 2005. Comments are more important than code. *ACM Queue* 3, 2 (2005), 64.