Astronomical Data Analysis Software and Systems VIII
ASP Conference Series, Vol. 172, 1999
D. M. Mehringer, R. L. Plante, and D. A. Roberts, eds.

A "Scientific" Approach to Software Project Management Part I: A Survey of Development Methodologies in Scientific Computing

Arno F. Granados

National Radio Astronomy Observatory[1]*, Green Bank, WV 24944-0002, USA*

Abstract. The author presents the motivation for creating a survey of software project management practices in scientific computing. The survey was distributed to attendees of ADASS'98, and the results and analysis will be presented at ADASS'99.

1. Motivation

I am a software engineer. I was an astronomer. My metamorphosis from astronomer to engineer occurred during a two year period when I worked for a small high-tech start up company creating commercial image processing software for remote sensing applications. Witnessing firsthand the differences between the process of creating "scientific" software, vs. writing "code-for-sale", made me question many aspects of software engineering, ranging from how many spaces to indent *if* ... *else* blocks, to project scheduling and workplace ethics. I have recently re-entered the world of astronomy, however this time I am working as an engineer, not a scientist. I have found that my background provides me with a unique perspective on the process of writing scientific software, and what I see from my perspective alarms me. Scientific software projects are becoming large and complex, even by industrial standards, and while scientists seem eager to borrow bleeding edge technology from industry (witness the proliferation of Java based tools at ADASS'98), they appear much less eager to borrow the methods used to manage these software projects.

I have been told that science and commercial programming practices don't mix, or that the situation is not as bad as I think it is. After hearing more than a modicum of contradictory information, I decided that it would be interesting and informative to ask scientists, scientific programmers, and software engineers working in science environments, just what exactly is going on in scientific software development. The amount of information was potentially overwhelming, so I decided to focus on the issues of software project management.

In the commercial-industrial world, Project Management (or lack thereof) and formal software development procedures are often cited for the success or failure of a software project (Whitten 1995). In contrast, scientific software has

[1]The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

© Copyright 1999 Astronomical Society of the Pacific. All rights reserved.

developed a reputation for being undisciplined, and formal development procedures are rare for even large projects–as an example, it wasn't until 1997 that the Science Software group at Space Telescope Science Institute implemented a formal program of design review and source code inspections (Shaw & Greenfield 1997).

Moreover, astronomical software is often "developed" by "Domain Experts", not software engineers. Historical inertia often binds institutions to a "code-and-fix" mind-set, amplified by the oft spoken observation that this is "research." Requirements are ill defined, if they exist at all. Yet, we all borrow software programs from colleagues, commission graduate students to write and maintain applications and scripts, utilize source code libraries (such as the IDL Astronomical Library), and invest resources in pseudo-commercial analysis packages such as AIPS, AIPS++, IRAF, and OPUS (Mandel & Murray 1998). While the computer industry has capitalized on the buzzword of "re-use," astronomers have been re-using code for decades. Unfortunately much of that code was originally written for single purpose applications, e.g., a single user, or as "throw-away" code, e.g., a single use. Such code is often associated with the words "unmaintainable" and "spaghetti-code." The ubiquitous computer has created the impetus for scientific software to be portable, robust, and maintainable (Conroy, Mandel, & Roll 1998): traits of software that has been designed and developed, not just coded and fixed.

2. The Survey

In order to analyze the current state of scientific software development in a more quantifiable way, I created a "Software Development Methodology Survey" which was distributed at the 1998 ADASS meeting. The purpose of the survey was to create a formal baseline of understanding of the software development methodologies in use in astronomical computing–what works and what doesn't. The survey was comprised of 33 questions covering the following topics:

- Are software groups creating formal requirements documents?
- How are developers obtaining requirements?
- Are software projects being tracked by project managers?
- Are developers using commercial tools (e.g., defect tracking, CASE tools, GUI tools), or building tools in-house?
- Do coding groups use techniques such as walk-throughs and code-reviews to limit code defects and improve reliability and maintainability?
- Are formal development procedures helpful or just too much overhead?
- What percentage of software groups have formal training in programming or computer science?

The survey was printed on multiple choice ScanTron forms. Statistics and analysis were not available by the December 15, 1998 deadline for proceedings submissions. The results and analysis of the survey will be presented at the 1999 meeting of ADASS.

Acknowledgments. I wish to thank the ADASS Organizing committee for their support and cooperation in distributing the survey. I also wish to thank

Kendra Bair at the University of Illinois for her assistance in preparing and processing the survey.

References

Conroy, M., Mandel, E., & Roll, J. 1998, in ASP Conf. Ser., Vol. 145, Astronomical Data Analysis Software and Systems VII, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 150

Mandel, E. & Murray, S. S. 1998, in ASP Conf. Ser., Vol. 145, Astronomical Data Analysis Software and Systems VII, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 142

Shaw, R. A. & Greenfield, P. 1997, in ASP Conf. Ser., Vol. 125, Astronomical Data Analysis Software and Systems VI, ed. G. Hunt & H. E. Payne (San Francisco: ASP), 26

Whitten, N. 1995, Managing Software Development Projects, (New York: Wiley)