# PAIR: A Rational Approach to Fighting Software Project Fires

Gregory T. Daich
*Software Technology Support Center*

*What happens when an organization finds a significant number of critical defects during testing? Whether the organization has a documented process or not, the organization naturally reverts to firefighting. It must quickly correct the problems before the schedule is "burned" beyond repair. The defined process for firefighting—which is likely called by other names, such as debugging, root cause analysis, or redesign or rework—may be followed. Regardless, it is project firefighting.*

## Red Adair

The book, *An American Hero: The Red Adair Story, An Authorized Biography* by Philip Singerman, tells of the famed firefighter who battles oil fires around the world. Revered as one of the most heroic firefighters, Adair has inspired us to assist organizations with immature document review practices to put out project fires. We have even seen some organizations with documented processes get caught firefighting too late in the project to overcome poor software quality.

## PAIR Service

The Software Technology Support Center's (STSC) Preliminary Analysis Inspection Report (PAIR) Service has demonstrated to many organizations an effective document review (inspection) process.[1] This service identified significant opportunities for improvement as well as the cause of the fire raging through many projects. We review a sampling of the organization's project documentation that is causing problems. The service then provides a report that can be used to initiate and plan improvements that focus on achieving desired quality levels.

## Typical Review Practices

After encountering a large number of defects, some people ask why the defects were not found in previous reviews. Document reviewers often are not given useful guidance in how to review documents. A manager merely slaps down a 100-page or more document and says, "We are going to have a meeting on this document in two days. I want you to review it."

The reviewer quickly skims through the document for the obvious problems or reads it late into the night to prepare for the meeting. Either way, many critical defects that could ignite project wildfires anytime during development are often missed.

Management usually has no problem signing off bug-infested documentation because project personnel cannot see the problems. More specifically, project personnel have not taken the time nor have they used effective techniques to inspect project documentation. It is not uncommon for organizations starting a new document inspection program to encounter 10 or more major defects per page in requirements specifications, designs, test plans, and process documentation.

## Enlightened Review Practices

A simple set of document rules and other useful tools and practices can turn a document skimmer or near comatose reviewer into an effective consultant who advises document authors about significant document issues. The PAIR concept initially joins the project organization with the STSC as partners in fighting fires caused by serious document defects. The document review practices we advocate help organizations dramatically improve their ability to find and remove many types of serious defects when it is cost-effective to do so, and ultimately to prevent them from occurring in the first place.

## Are CMM[2] and the J-STD-016[3] Enough?

Using the right tools, it does not take much effort to determine that a project has document quality problems. Many organizations have conducted process improvement initiatives, which started with a Capability Maturity Model-flavored process assessment. However, the guidelines for conducting these assessments are mainly concerned about document existence and not the quality of the project documentation. Some process assessors have reviewed a few document samples during an assessment and found useful information about process maturity.

I have seen approved software test plans that are void of test planning information. I have even seen software development plans that did not contain schedule, task, and product deliverable information. The main reason these significant problems exist is often because people do not fully understand the purpose of the required documents. Standards like the J-STD-016 should help us write better documents, but without effective and efficient review practices, the standards and guidelines do little.

Institute of Electrical and Electronics Engineers (IEEE) 1028, Software Reviews, provides some useful generic review practices that are worth considering. But this standard will still need to be customized for an organization's specific needs.

## PAIR Service Demonstrated

Projects that are experiencing significant quality and testing problems can be assessed to determine if document quality is significantly inhibiting software quality and testing effectiveness. Understanding the purpose and objectives each document is

supposed to accomplish and mapping the contents against those objectives often finds gaping holes in the document's content. A no-cost demonstration of our PAIR service has substantiated this fact for many organizations. It is a simple yet powerful approach to reviewing that changes how people look at documents and can radically improve overall project performance and software quality.

We will use our PAIR Service to help you extinguish fires that threaten your projects. Recovering from poorly written requirements documents during systems testing will be expensive compared to recovering during the requirements phase. However, it will be much less expensive than recovering after system delivery. PAIR will help you start a document quality improvement initiative before the fires rage and help you achieve your project goals.

## Conclusion

We are not advocating firefighting as a way of life for software developers. We advocate a rational approach to stamp out fires when they occur. A consequence of this approach is the ability to prevent project fires and many types of software quality problems in the first place.

We do not need heroes who remove defects just before delivery after they inserted them throughout development. Red Adair does not start the oil fires he is asked to extinguish, but we need heroes like him in the software world who can put them out before we lose everything. We need heroes to identify significant issues early, helping authors improve document quality and succeed in meeting document and project objectives. ◆

## About the Author

Gregory T. Daich is a senior software engineer with Science Applications International Corporation currently under contract with the Software Technology Support Center. He supports STSC's Software Quality and Test Group with more than 22 years experience in developing and testing software. He has taught more than 60 software test, document inspection, and process improvement seminars in the last five years.

Daich consults with government and commercial organizations on improving the effectiveness and efficiency of software quality practices. His consulting approach coordinates formal document inspections with analysis of test work products to identify opportunities for software test process improvement. These practices have also been applied in supporting and testing year 2000 upgrades. He is the principal author for several guidebooks and workshops for conducting year 2000 compliance projects. These guidebooks address corporate- and project-level compliance efforts as well as year 2000 desktop (PC and Macintosh) software compliance.

Daich has a master's degree in computer science from the University of Utah and a bachelor's degree in mathematics from Weber State University.

Software Technology Support Center
7278 Fourth Street
Hill AFB, UT 84056-5205
Voice: 801-777-7172
Fax: 801-777-8069
E-mail: daichg@software.hill.af.mil
Internet: http://www.saic.com

## Notes

1. For example, see Tom Gilb's book *Software Inspections*, Addison-Wesley, 1993.
2. CMM stands for the Capability Maturity Model developed by the Software Engineering Institute with the support of many government and commercial organizations.
3. J-STD-016 is the Electronic Industries Association Institute of Electrical and Electronics Engineers Standard for Information Technology Software Life Cycle Processes Software Development Acquirer-Supplier Agreement, 1995.