

# Are You Ready to Deliver? To Ship? To Test?

Capt. Brian Hermann, *U.S. Air Force*  
Jim Russell, *Honeywell, Inc.*

*How do you know when you are ready to deliver your software product? Do you ship on your contract delivery date? Are you under pressure to get the product to market and beat the competition? What kind of measures and metrics do you use to make the decision to deliver? This article introduces the method the Air Force Operational Test and Evaluation Center uses to determine whether a system's software is mature enough for the system to enter dedicated operational test and evaluation and to subsequently be fielded or procured. Though we use this evaluation to determine test readiness and suitability for fielding and procurement, it could easily be applied as an exit criteria for software development or as a component of the "decision to ship" process.*

**S**oftware product maturity: a measure of the progress software products are making toward satisfying user requirements [1].

The Air Force Operational Test and Evaluation Center (AFOTEC) uses a software product maturity evaluation as a test readiness criterion for a system's software prior to a system's entry into dedicated operational test and evaluation (OT&E). Our evaluation is a trend analysis of software changes identified during software development and testing. Based upon this analysis, AFOTEC provides a recommendation of the software's readiness for operational testing. This evaluation currently does not provide any kind of projection of future software product maturity but provides an excellent snapshot of system maturity. (However, AFOTEC has performed some investigation into methods of maturity projection [2].) The entire method is documented in [1].

## Software Product Maturity Data Requirements

Our experience shows that most developers and procurement offices already collect the data necessary to perform a software product maturity evaluation. Data that describes and tracks documented software changes serves as the key input to the evaluation. Following are the minimum data required for each software change to evaluate software product maturity.

- Software change (problem) number.
- Description.
- Computer Software Configuration Item (CSCI) Identifier.
- Severity level.
- Date change opened (or problem found).
- Date change (problem) closed and implemented.

For the change Severity Level definitions, AFOTEC adapted the "Priority Classifications for Problem Reporting," listed in Appendix C of MIL-STD-498. Using these definitions, systems with open Severity Level 1 or 2 software changes are not recommended for entry into dedicated OT&E. Many organizations use different severity, criticality, or priority definitions. Any reasonable ranking system is acceptable as long as a clear definition of product maturity is included.

## Software Changes

By software *change*, we mean any change that

- Corrects errors (corrective change).

- Enhances system capability (perfective change).
- Makes the software compatible with changes in the computing environment (adaptive change).

In our evaluations, we include software problem reports, software failure reports, software change requests, trouble reports, and any other data that fits the above definitions. If the software does not meet user requirements, the documentation of the unmet need is an input for the software product maturity evaluation.

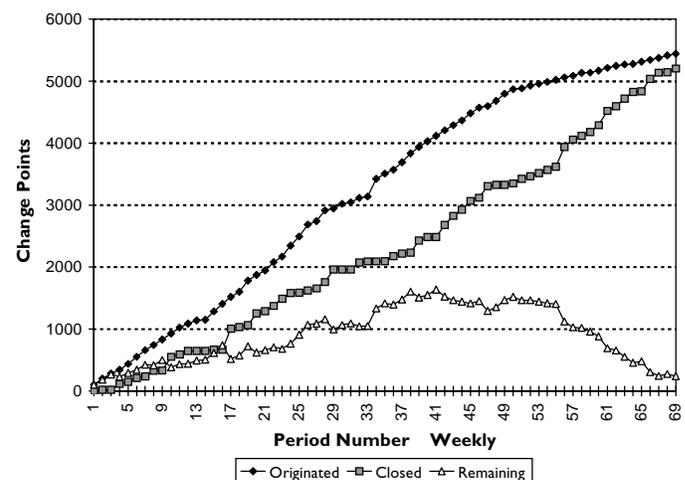
## External Factors

To correctly gauge readiness to deliver, developers must also evaluate test completeness, test rates, and requirements stability. Any of these factors can cause product maturity to look unrealistically good or bad. Obviously, if only 10 percent of the planned tests have been completed, it is premature to ship the product—despite low software change trends. Likewise, high test rates will likely produce more changes and problems than lower test rates. Requirements instability is one of the most common causes of software product immaturity of the Department of Defense's long development cycle projects.

## Maturity Evaluation and Analysis Tool

AFOTEC developed a Microsoft® Excel for Windows™-based tool, called Maturity Evaluation and Analysis Tool, to

Figure 1. Accumulated software changes (weighted).



automate the data manipulation, produce trend charts, and speed analysis and reporting. The tool and user's manual are available to U.S. government offices and their contractors at no cost from HQ AFOTEC's Software Analysis Division at DSN 246-5310 or E-mail sas@afotec.af.mil.

### Trend Charts and Analysis

Software product maturity evaluation entails a graphical analysis of change data trends in the context of project schedule and other external factors. The basic product maturity chart (Figure 1) shows the *total changes originated, closed, and remaining* trends. (Note: These charts contain data from multiple, real systems and are provided as examples only.) To indicate maturity or progress toward maturity, the total changes originated trend should begin to level off. This indicates testing is finding problems at a decreasing rate. If problems are being closed efficiently, the total changes closed curve should closely follow the total originated trend. Ideally, all identified changes are closed, and the remaining changes curve would show no backlog.

Although the *remaining changes* trend in the basic chart shows the current software problem or change backlog, Figure 2 presents a more useful view. This stacked bar chart shows the overall backlog trend as well as each severity level's contribution to the total backlog.

Figure 3 shows both remaining changes for each CSCI and the *defect density* (the number of remaining changes or problems divided by thousands of new or modified source lines of code). In addition to the minimum change data, defect density analysis requires code size information. Literature suggests software is not ready for release until the defect density is below 0.5 [3]. Rather than blindly endorse this number, we suggest developers select a threshold of their own. Finding portions of software with the most remaining problems and the highest defect densities are two additional pieces to the product maturity puzzle.

Our product maturity tool produces over a dozen additional trend charts including average severity, severity level

Figure 2. Remaining software problems (unweighted).

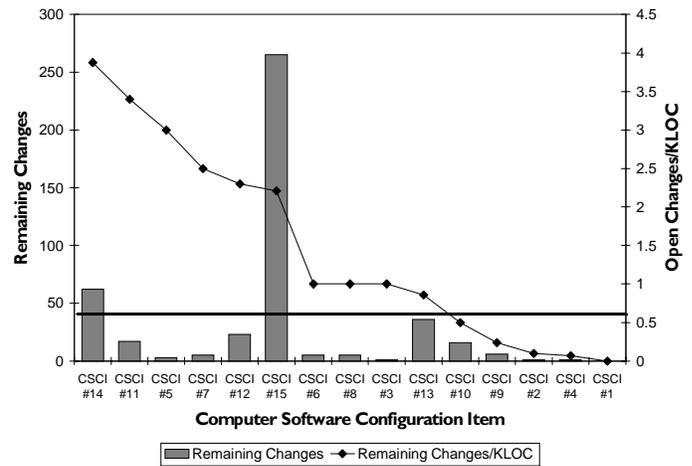
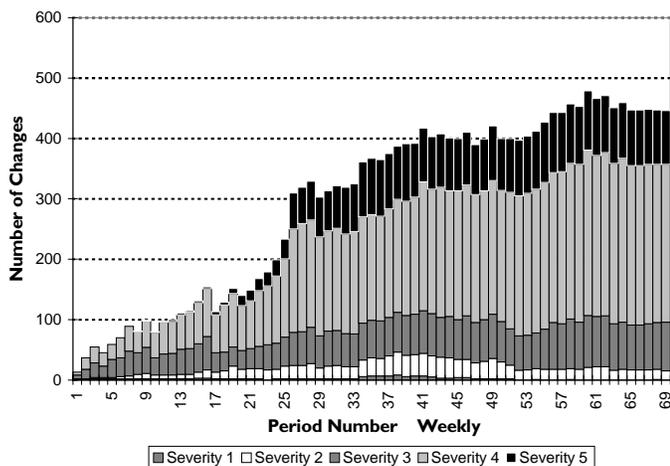


Figure 3. Remaining changes and defect density.

distribution, average closure time, charts for each severity level, and charts for each configuration item or subsystem.

### Value to Software Developers and Software Professionals

The AFOTEC approach for evaluating software product maturity is directly transferable to any software development activity and has much more value to a software developer than to an operational tester. Table 1 (page 30) shows just a few possible uses of the software product maturity evaluation.

### Conclusion

Some of you may be thinking “so what.” Yes it is true—this data is usually collected and readily available. Beyond managing rework, however, few developers take full advantage of this data housed in their own configuration management systems. Developers, as owners of the data required to perform a software product maturity evaluation, are best able to perform this evaluation and use the results to improve the quality of their products. Having metrics to back up your answer to the “are you ready ...” question will help base your decision on quantitative facts vs. reliance on your gut feelings.

As former software maintainers, we wish we had been able to see the snapshot this maturity evaluation provides. It has been said that a “picture is worth a thousand words.” What is the picture of your software product's maturity worth to you? ♦

### About the Authors



**Capt. Brian Hermann**, U.S. Air Force, is an Air Force Institute of Technology (AFIT) computer science doctoral student at Arizona State University in Tempe, Ariz. His current area of study is software engineering. He was previously assigned to the AFOTEC at Kirtland Air Force Base, N.M. He has a master's degree in software systems management from AFIT and a bachelor's degree in electrical engineering (computer concentration) from the University of Notre Dame.

See *READY TO DELIVER*, page 30

READY TO DELIVER from page 15

Possible Uses	Explanation
Maintenance Effort Estimation	Use change rate and rework effort information to estimate required maintenance resources.
Process Improvement	Track common causes of changes to identify process changes that might eliminate rework.
Project Management	Focus management interest and development effort on configuration items, subsystems, or feature areas with high numbers of total or remaining changes.
Delivery, Ship, or Release Decision	Set readiness criteria, e.g., no remaining high-severity problems, maximum number of remaining problems, or maximum defect density, prior to product delivery.
Rework Management	Prioritize software changes according to severity and customer priorities.
Schedule Prediction	Use recent closure and identification rate trends to estimate when the software product will meet test or release criteria.
Test Readiness	Set readiness criteria, e.g., no remaining high-severity problems, maximum number of remaining problems, or maximum defect density, prior to field testing.

Table 1. *Software product maturity uses.*

AFIT/CIGW  
 c/o AFOTC Det. 025  
 Arizona State University  
 Tempe, AZ 85287  
 Voice: 602-704-0135  
 Fax: 602-704-0135  
 E-mail: brian.hermann@asu.edu  
 Internet: <http://www.public.asu.edu/~bgh14/maturity>



**Jim Russell**, a former Air Force captain, served as chief of Software Evaluation Methods, Education, and Automation, Software Analysis Division, AFOTEC at Kirtland Air Force Base, N.M. He was responsible for evaluating Air Force systems for software maintainability, supportability, and maturity. He also led the team that improved current evaluations, researched new evaluation methods, coordinated software training requirements, and headed the office's automated software evaluation efforts.

Russell is a graduate of the Air Force Software Professional Development Program at AFIT, and has a bachelor's degree in computer science from Loyola Marymount University. He is currently working toward a master's degree in engineering management from the University of Colorado.

Russell currently works for Honeywell, Inc. in Phoenix, Ariz.

HQ AFOTEC/SAS  
 8500 Gibson Boulevard SE  
 Kirtland AFB, NM 87117-5558

Voice: 505-846-5310 DSN 246-5310  
 Fax: 505-845-5145 DSN 246-5145  
 E-mail: sas@afotec.af.mil or jim.russell@acm.org  
 Internet: <http://www.afotec.af.mil>

## References

- Hermann, Brian G., AFOTEC Pamphlet 99-102, Vol. 6, "Software Maturity Evaluation Guide," March 1, 1996.
- Goel, Amrit L. and Brian G. Hermann, "Software Maturity Evaluation: When Is Software Ready for Operational Testing or Fielding?" presented at the Software Technology Conference, Salt Lake City, Utah, April 1997.
- Foody, Michael A., "When is Software Ready For Release?" *UNIX Review*, March 1995.

## Coming Events

### '98 Software Engineering Institute Symposium

**Dates:** Sept. 14-17, 1998

**Location:** Pittsburgh, Pa.

**Subject:** This symposium provides a forum to articulate currently applicable practices that software practitioners can use to improve *what* they build by improving *how* they build.

**Sponsor:** Software Engineering Institute

**Contact:** Voice: 412-268-5800

**E-mail:** [customer-relations@sei.cmu.edu](mailto:customer-relations@sei.cmu.edu)

**Internet:** <http://www.sei.cmu.edu>

### Call for Participation: Thirteenth International Forum on COCOMO (COConstructive COSt MOde) and Software Cost Modeling

**Theme:** Software Sizing

**Dates:** Oct. 6-8, 1998

**Location:** Los Angeles, Calif.

**Subject:** This year's forum particularly solicits presentations on software sizing, e.g., object points or other graphical measures, function points, UML-based or other object-oriented measures, alternative sizing techniques, comparison of software size measures, and usage and calibration of size measures in the COCOMO II submodels and other cost models.

**Sponsor:** University of Southern California Center for Software Engineering and the Software Engineering Institute

**Contact:** Jennifer Browning, Center for Software Engineering, University of Southern California, Los Angeles, CA 90089-0781

**Voice and Fax:** 213-740-5703

**E-mail:** [browning@sunset.usc.edu](mailto:browning@sunset.usc.edu)