

## CROSSTALK would like to thank 309 SMXG for sponsoring this issue.



How would you define success? On a personal level, you may state that success is the attainment of happiness or the completion of an effort you were passionate about. In development, success might be defined by a set of criteria against stated objectives. And yet, there is ambiguity in exactly what success is here as well. For example, from a

developer's point of view, it may be the system's technical capability. From a user's perspective, success is generally defined by key performance and functional attributes. A customer's considerations may be focused toward budget, schedule, quality, and user satisfaction. Are success criteria absolute in nature? Do we automatically consider a program to have failed if it does not achieve all stated objectives? Or, should success be measured in by increments, depending on the observer's point of view?

As we turn the page on another year and move steadfastly into the next, we here at **CROSSTALK** have chosen to dedicate this issue to looking retrospectively on the evolution of the software engineering field. The intent is not to simply reflect on the past and cast black-and-white judgments on whether something was a 'hit' or a 'miss,' but rather to delve deeper into the nature of success and what causes the perception of failure to begin with. Truly, some ideas we once believed were revolutionary truly were. But in hindsight, others missed the mark despite our best efforts. The question is, from what point of view did they succeed, and for whom did they fail?

Our desire to simplify topics to black-and-white criteria speaks to the limited perspective and fallibility of mankind, especially in realms with ever-increasing complexity. We are quick to label something a success or a failure, though the truth is most often somewhere in between. One truism that we can count on is that failure is inevitable; how we cope with that failure and progress beyond it defines who we are. It was Robert F. Kennedy in his 1966 Day of Affirmation address who stated, "We also know that only those who dare to fail greatly, can ever achieve greatly."

Although the emphasis here is on the potential benefits weighed against calculated risks, it is important to note that not all missteps are necessarily failures — just fundamental steps in the nature of progress. The culmination of our combined successes and failures, along with our ability to learn from them, enhance possibilities and drive technology forward. Remembering that success is relative to your perspective, however, is imperative to knowing not just what went wrong, but also what went right.

It is with that thought that we begin this issue of **CROSSTALK** with Dr. David Cook and Dr. Eugene Bingue's article entitled "Ada — A Failure That Never Happened." In this

article, we explore the external factors, such as the exponential proliferation of programming languages, that led to problems in maintenance and continued support. Continuing with the topic of Ada, we also have a fine article on the topic by Dr. Drew Hamilton and Dr. Patrick Pape entitled "Ada — 20 Years After The Mandate" that takes a retrospective look at the Ada mandate and is certain to be enlightening from the perspective of an author who is directly linked to the Ada Joint Program Office (AJPO).

We have two articles on the topic of Agile. Dick Carlson has written an insightful piece entitled "Scrum is Simple" that discusses the cultural shift away from traditional project management methods and tackles the difficulties of changing the 'status quo' and implementing Agile for projects looking to transition to Agile methodologies. The second piece on Agile is entitled "Why Agile Projects Sometimes Fail" by Gerald Weinberg. This is an open forum piece that discusses the effective leadership styles and team skills necessary for a productive Agile project.

In this special issue, we also feature two articles by Capers Jones. The first piece, entitled "A Comparison of Medical Diagnoses and Software Problem Diagnoses," illustrates the need for a higher degree of rigor in empirical data by comparing and contrasting software to the medical field. The second article, entitled "A Retrospective View of the Laws of Software Engineering," takes laws and observations from other scientific fields and shows the applications and truisms found within each and how they apply to the software engineering realm as well.

For those of you who wish to have an increasingly technical read, we have chosen to include a special supporting article entitled "Generating Actionable Information for Classifier Assessment" by E. Earl Eiland and Dr. Lorie M. Liebrock. This article will arm practitioners with the means to calculate actionable information for their specific mission. Finally, do not forget to read our ever-insightful "BackTalk" column, written by Dr. David Cook. In his piece, entitled "Failure IS an Option," he brings us full circle to the nature of success and failure and what we can learn from both.

As we begin the new year, we are also beginning the 29th year of **CROSSTALK**, would like to express my sincere thanks to everyone for making such an accomplishment possible. We thank our co-sponsors for your generous support and active involvement in providing an informational and educational resource to the software industry. To the authors, we truly appreciate all of your time and effort in sharing such valuable information with the software community. To our readers, thank you for your continued support. We hope that we continue to exceed expectations by publishing the highest-quality articles.

**From all of us at CROSSTALK, we wish you the best for the new year.**

**Tracy Stauder**  
Deputy Director, 309 Software Maintenance Group