

Myths and Strategies of Defect Causal Analysis

David N. Card
Q-Labs, Inc.
dca@q-labs.com

Biography

David N. Card is a fellow of Q-Labs, a subsidiary of Det Norske Veritas. Previous employers include the Software Productivity Consortium, Computer Sciences Corporation, Lockheed Martin, and Litton Bionetics. He spent one year as a Resident Affiliate at the Software Engineering Institute and seven years as a member of the NASA Software Engineering Laboratory research team. Mr. Card is the author of *Measuring Software Design Quality* (Prentice Hall, 1990), co-author of *Practical Software Measurement* (Addison Wesley, 2002), and co-editor *ISO/IEC Standard 15939: Software Measurement Process* (International Organization for Standardization, 2002). Mr. Card also serves as Editor-in-Chief of the *Journal of Systems and Software*. He is a Senior Member of the American Society for Quality.

Abstract

The popular process improvement approaches (e.g., Six Sigma, CMMI, and Lean) all incorporate causal analysis activities. While the techniques used in causal analysis are well known, the concept of causality itself often is misunderstood and misapplied. The article explores the common misunderstandings and suggests some strategies for applying causal analysis more effectively. It does not provide a tutorial on any specific causal analysis technique.

Many different processes, tools, and techniques (e.g., Failure Mode Effects Analysis, Ishikawa diagrams, Pareto charts) have been developed for defect causal analysis. All of them have proven to be successful in some situations. Organizations often invest large amounts in the software and training needed to deploy them. While application of these techniques helps gain insight into the sources of problems, a checklist implementation of the techniques alone is not sufficient to ensure accurate identification and effective resolution of “deep” problems. Gaining a true understanding of concepts underlying causality and developing a strategy for applying causal analysis help to maximize the benefit of an organization’s investment in the tools and techniques of causal analysis.

Introduction

Causal analysis is a major component of modern process improvement approaches, e.g., the CMMI, Six Sigma, and Lean. The implementation of effective causal analysis methods has become increasingly important as more software organizations transition to higher levels of process maturity where causal analysis is a required as well as appropriate behavior. Petrosky [1] argues that failure, and learning from failure, is an essential part of engineering discipline. He describes how the investigation of some spectacular failures has led to improvements in scientific knowledge and engineering practice. However, software engineering organizations often are reluctant to acknowledge failure and often don’t act effectively to investigate it.

While many things can go wrong in the implementation of any new tool or technique, my experience suggests that four myths about the nature of causal analysis are common impediments to its success in software engineering:

- Causality is intuitive, so it needs no explanation
- Causal analysis can be done effectively by external tiger teams or graybeards
- Causal analysis is for “mature” organizations only
- Small improvements don’t matter

Resolving these misconceptions and developing a strategy to minimize their effects helps to maximize the benefits of this simple, but important learning activity.

While causal analysis of some form can be applied to almost any type of anomaly, defects are the most common subject of causal analysis. Thus, this article will focus on defect causal analysis, although these issues apply to investigations of other types of anomalies, as well. Before discussing the myths, let's review some basic concepts.

Concept of Causal Analysis

Causal analysis focuses on understanding cause-effect relationships. A causal system is an interacting set of events and conditions that produces recognizable consequences. Causal analysis is the systematic investigation of a causal system in order to identify actions that influence a causal system, usually in order to minimize undesirable consequences. Causal analysis may sometimes be referred to as root cause analysis or defect prevention.

Many good examples of causal analysis efforts in software engineering have been published, e.g., [2], [3], [4], [5], and [6]. However, these efforts have adopted different terminology and approaches. (Card [11] proposes a unifying set of terms and concepts for describing causal analysis.) The differences between the analysis procedures obscure the commonality in the subject matter to which the procedures are applied. Further complicating the situation are apparent differences in the notion of causal analysis defined in the CMM [7] and CMMI [8], see Card [11]. This cloud of terminology creates an environment in which myths can thrive.

Myth 1: Causality is Intuitive

Causal analysis focuses on understanding cause-effect relationships. Searching for the cause of a problem is a common human endeavor that wouldn't seem to require much formalism. However, causal investigations often go wrong from the beginning because the investigators don't recognize what truly constitutes a "cause". Three conditions must be established in order to demonstrate a causal relationship:

- First, there must be a correlation or association between the hypothesized cause and effect
- Second, the cause must precede the effect in time
- Third, the mechanism linking the cause to the effect must be identified

The first condition implies that when the cause occurs, the effect is also likely to be observed. Often, this is demonstrated through statistical correlation and regression.

While the second condition seems obvious, a common mistake in the practice of causal analysis is to suppose that a cause-effect relationship exists between factors that occur simultaneously. This is an over-interpretation of correlational analysis. Consider the situation in which the time spent by reviewers preparing (reviewing) and the number of defects found while reviewing are recorded from peer reviews. (A separate meeting may be held to consolidate findings from multiple reviewers.) Figure 1 shows a scatter diagram of these two measures of peer review performance. These two variables frequently demonstrate significant correlations, thus satisfying Condition 1 (above). This diagram and a correlation coefficient computed from the data often are taken as evidence of a causal relationship between review and detection.



Figure 1. Example of Correlation Between Variables

However, most defects are discovered *during* the individual review time. A few more may be identified during a consolidation meeting. Both meters (review time and defects) are running simultaneously – one does not occur before the other, thus Condition 2 (above) is not satisfied. There is a necessary relationship between review time and defect detection, but not a causal one. The difference is important.

Defects cannot be found unless review time is spent – that’s necessary. However, it is not necessarily true that simply directing reviewers to spend more time will cause more defects to be found. If the reviewer has already read the material once, reading it again (the same way) isn’t likely to produce new findings. Some improvement to the reading technique is needed. If the reviewer didn’t have time to finish reading the material, then a mandate to spend more time is likely to be ineffective, unless something is done to provide time in the reviewer’s schedule. In each case a focus on increasing review time short-circuits the causal analysis and leads to an ineffective action. Instead, this correlation should suggest that some other factor that affects both review time and defect detection should be sought.

Issuing a mandate (as a corrective action) to spend more time reviewing may result in more time being charged to reviews (to show compliance), but it isn’t likely to increase the defects detected. The underlying cause of low review time and defect detection may be a lack of understanding of how to prepare, schedule pressure, or other factors that affect both measures. That underlying cause must be addressed to increase both the review time and defect detection. Recognition of the correlational relationship helps to narrow the set of potential causes to things that affect both measures. Simply encouraging reviewers to spend more time reviewing is only likely to improve defect detection if the underlying problem is laziness or lack of concern for quality. (That is often management’s perception.) However, if the reviewers are making a good faith effort to follow the defined process then this encouragement isn’t helpful.

Some of the responsibility for this kind of misinterpretation of correlational relationships can be attributed to statisticians. The horizontal and vertical axes of Figure 1 are typically referred to as the “independent” and “dependent” variables

respectively. While these terms are simple labels, not intended to imply a causal relationship, they are often misunderstood. Applying the three criteria for causality helps to ensure that the causal analysis team probes beyond the obvious correlation.

Because causal analysis is perceived to be “intuitive”, many organizations do not provide meaningful training in it to their staff. Even Six Sigma training programs, which often devote a lot of time to causal analysis techniques, typically do not provide an explicit definition of causality such as the three conditions for causality discussed above. In the course of my work with clients I have reviewed the content of several Six Sigma Black Belt training programs. None of those I examined provided a definition of causality or of a causal system. Try to find one in your training!

One of the consequences of a poor understanding of the nature of causality is that causal analysis sessions become superficial exercises that don’t look deeply enough to find the important causes and potential actions that offer real leverage in changing performance. This reduces the cost-benefit of the investment in causal analysis expected of mature software organizations. Causal analysis training should address the underlying concepts, not just the forms and checklists that make up the techniques.

Myth 2: Causal Analysis by “Tiger Team”

Many organizations implement causal analysis by assigning “tiger teams” or “graybeards” drawn from outside the process or project to investigate the anomalies. The idea is that smart and experienced staff can solve problems created by less experienced and skilled staff. This reflects an “audit” mentality. The assumption is that problems occur because the staff fails to adopt correct behaviors, not because the staff is working in an ambiguous and changing environment.

It is informative to look at the context in which the Lean approach positions causal analysis in manufacturing situations. The Japanese term “gemba” may be translated descriptively as, “the scene of the crime”. The idea is that causal analysis should take place in the actual location (gemba) where the defect occurred. There may be factors in the environment, for example, that contribute to the problem that might not be apparent in a conference room far away from the factory floor. The most effective causal analysis will consider all influences on the problem.

The software development process is an intellectual process. It takes place in people’s heads. The people executing the software process need to be involved in the causal analysis process. The method recommended by Card [6] and others with practical experience put the responsibility for defect causal analysis on the software developers or maintainers who contribute to the mistakes. They are the best qualified to identify what went wrong and how to prevent it.

Some Six Sigma implementations exacerbate this problem by systematically assigning the responsibility for causal analysis to “black belts” who may have little experience with the specific process under study or the defects being investigated. No amount of training can substitute for the insight obtained by actually being involved in the situation.

Myth 3: Causal Analysis for High Maturity Only

The CMMI assigns Causal Analysis and Resolution to Maturity Level 5. Similarly, its predecessor, the CMM assigned Defect Prevention to Maturity Level 5. This gives the impression that only mature organizations can benefit from formal causal analysis. Six Sigma programs often embed causal analysis in a curriculum of much more challenging statistical techniques, discouraging the independent application of simple defect causal analysis techniques.

Causal analysis can be applied effectively to any well-defined process. This doesn’t require the entire process to be well-defined, only the parts to which causal analysis are applied. Moreover, the process definition needs to be complete, but doesn’t need to satisfy any specific CMM/CMMI maturity or capability level. Dangerfield, et al. [3] describe the successful application of causal analysis within a CMM Level 3 project. Card [6] describes the application of Defect Causal Analysis to an “independent” configuration management organization, with little software development activity.

Myth 4: Small Improvements Don’t Matter

Human nature tends to favor revolutionary and dramatic change over incremental and sustained improvement. Many of the recommendations resulting from causal analysis teams are small changes that improve communications and processes in small ways. However, these small changes can have dramatic impacts over the long-term.

Consider the real example (described in [3]) of a project that identified “inconsistent use of the computing environment” as a systematic problem and acted to eliminate it by standardizing on naming, security, and interface conventions. These actions “cleaned up” some aspects of the project’s process. Figure 2 shows the reduction in defect rate from Build 1 to Build 2 as a result of these actions. Clearly, these small changes had a big impact!

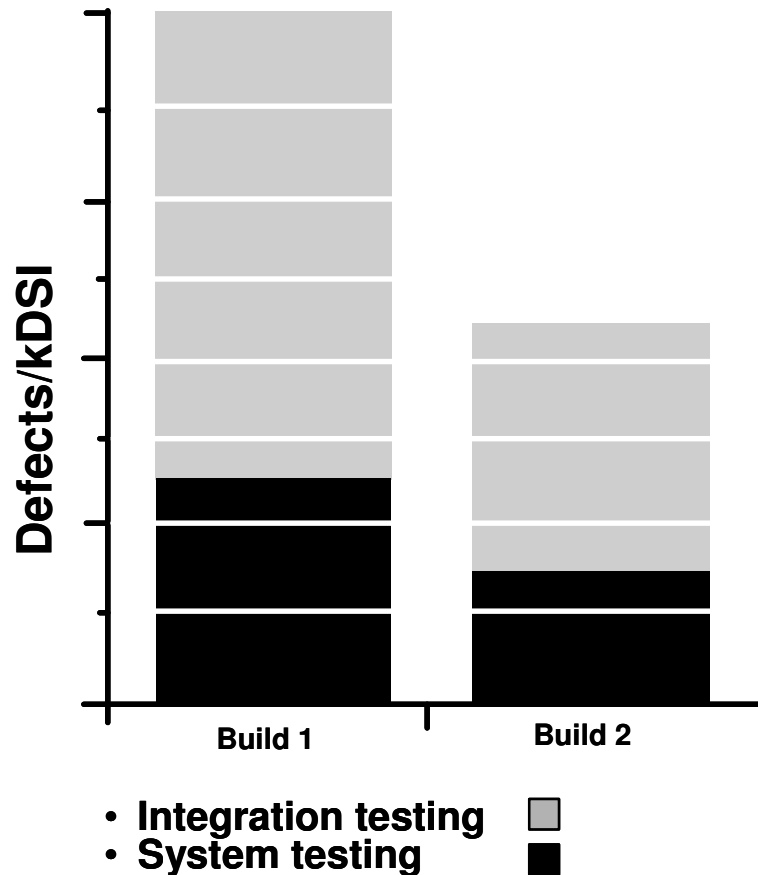


Figure 2. Example Effect of Incremental Improvement (from [3])

While many organizations might prefer to take more dramatic actions (e.g., new technology and processes) with a correspondingly larger investment, substantial improvements often can be obtained with incremental changes to current processes.

Strategies for Causal Analysis

Often, the causal analysis process and techniques are taught to staff, but little guidance is provided on when and where to apply them. Consequently, the application of causal analysis becomes ad-hoc. Moreover, management often assumes that once people are trained, they will somehow identify an important problem and work the causal analysis task into their schedule. A causal analysis strategy should define when and where causal analysis should be performed. It is based on an understanding of the organization’s process improvement objectives as well as its current performance levels. The causal analysis strategy helps to ensure that resources are applied systematically to important problems within the organization.

A causal analysis strategy should address the following elements of the causal analysis program:

- Training – explain the concepts, techniques, and context for causal analysis. Make the training widely available. Anyone in the organization could be a candidate for some causal team. Training should include the following:
 - Definition of causality and causal systems.

- Causal analysis techniques and tools. Limit training “for the masses” to the few most appropriate techniques
- Description of the organizational context for planning and performing causal analysis, e.g., the strategy
- Focus – identification of key problem areas to be worked. Of course, some causal analysis may be triggered by “special causes”, but don’t just wait for a “black belt” to volunteer to work on an important and persistent problem. Establishing a focus requires the following:
 - Identify the production activities that are the greatest sources of defects
 - Identify the testing activities that are least effective
 - Establish causal analysis teams for the activities that represent the greatest improvement opportunity
 - Put effective data collection mechanisms in place for these activities
 - Schedule causal analysis of these activities at the appropriate point in the project life cycle
 - Allocate resources for performing causal analysis
- Action – implement the recommendations of the causal analysis teams. The benefits of causal analysis are lost without timely action. Ensuring that action is taken requires the following:
 - Establish an action team that includes management and technical experts
 - Schedule regular reviews of causal analysis activities and results
 - Allocate resources to implementing causal analysis team recommendations
- Communication – keep staff informed about the status of causal analysis activities and the lessons learned from the investigations.
 - If causal analysis team members don’t realize that their recommendations are being acted upon, then they lose interest in the process
 - If the larger staff isn’t informed of the outcome, they can’t learn from the lessons accumulated

Many of these elements are addressed in the Defection Prevention process area of the CMM, but have been dropped from Causal Analysis and Resolution in the CMMI (see Card [11]). Unless these four elements of strategy are addressed, the causal analysis program will remain ad-hoc and perform fitfully.

Summary

Defect causal analysis is becoming ever more common in the software industry as process maturity increases and new forces, such as Six Sigma and Lean, focus increasing attention on quality improvement. The four myths previously discussed often impede the effective implementation of a causal analysis process. While causal analysis seems “obvious”, meaningful training and thoughtful planning are needed to ensure its successful implementation. This article has attempted to identify some of the major problems and suggest solutions to them.

References

- [1] Petrosky, H., *To Engineer is Human; The Role of Failure in Successful Design*, St. Martin’s Press, 1985
- [2] Mays, R., et al., Experiences with Defect Prevention, *IBM Systems Journal*, January 1990
- [3] Dangerfield, O., et al., Defect Causal Analysis - A Report from the Field, *ASQC International Conference on Software Quality*, October 1992
- [4] Yu, W., A Software Fault Prevention Approach in Coding and Root Cause Analysis, *Bell Labs Technical Journal*, April 1998
- [5] Leszak, M., et al., Classification and Evaluation of Defects in a Project Perspective, *Journal of Systems and Software*, April 2002
- [6] Card, D., Learning from Our Mistakes with Defect Causal Analysis, *IEEE Software*, January 1998
- [7] Paulk, Mark, et al., *Capability Maturity Model*, Addison Wesley, 1994
- [8] CMMI Development Team, *Capability Maturity Model – Integrated, Version 1.1*, Software Engineering Institute, 2001
- [9] Ishikawa, K., *Guide to Quality Control*, Asian Productivity Organization Press, 1986
- [10] Harry, M. and R. Schroeder, *Six Sigma*, Doubleday, 2000
- [11] Card, D., Understanding Causal Systems, *Crosstalk*, October 2004