Understanding Causal Systems

David N. Card
Software Productivity Consortium

This article describes a model and a supporting set of terms that facilitate reasoning about causal systems, planning for causal analysis, and designing process experiments. This perspective is based on practical experience in implementing causal analysis in industry. 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 an appropriate – behavior.

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 to minimize undesirable consequences. Causal analysis may sometimes be referred to as root cause analysis or defect prevention. Searching for the cause of a problem (*laying the blame*) is a common human behavior that would not seem to require much formalism. However, causal investigations often go wrong, beginning with the definition of a cause.

Causal analysis focuses on understanding cause-effect relationships. Three conditions must be established 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 hypothesize cause-effect relationships between factors that occur simultaneously. This is an over-interpretation of the correlational analysis.

Figure 1 shows a scatter diagram of two variables measuring inspection (or peer review) performance. These two variables frequently demonstrate significant correlations. This diagram and a correlation coefficient computed from the data often are taken as evidence that preparation causes detection.

However, most inspection defects are discovered *during* preparation. Both meters are running simultaneously. Thus, preparation performance cannot substantially influence detection performance. They are measures of the same activity. Rather, the correlation suggests that some other factor affects both preparation and detection.

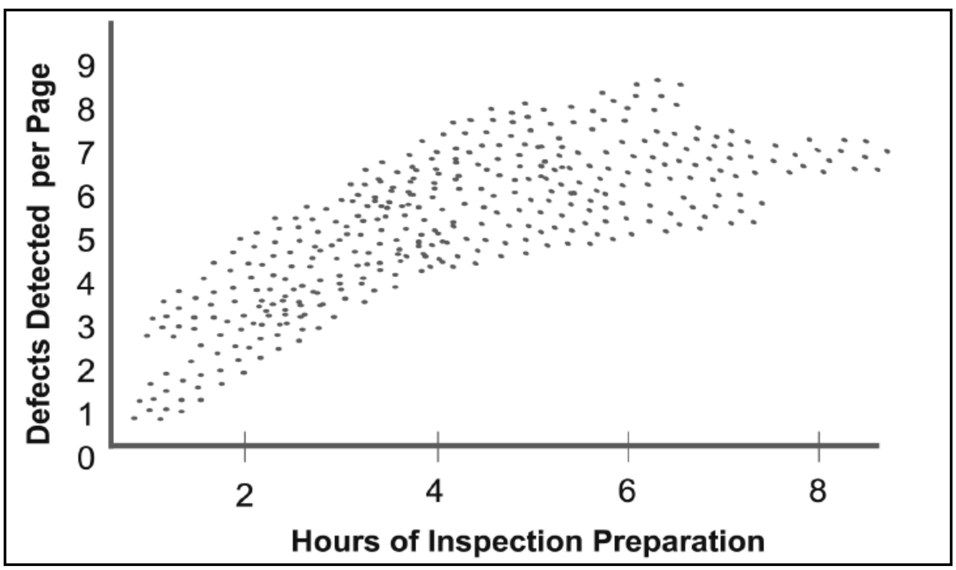
Issuing a mandate (as a corrective action) to spend more time in preparation may result in more time being charged to inspections, but it is not likely to increase the defect detection rate. The underlying cause of low preparation and detection rates may be a lack of understanding of how to prepare, schedule pressure, or other factors that affect both measures. That underlying cause must be addressed to increase both the preparation rate and detection rate. Recognition of the correlational relationship helps to narrow the set of potential causes to things that affect both preparation and detection performance.

The relationship between the height and weight of adult human beings provides a good analogy to the situation described in Figure 1. Taller people tend to weigh more than shorter people. (Obviously other factors intervene as well.) While this is a necessary relationship, it is not a causal relationship. It would be a mistake to assume that increasing someone's weight would also increase his/her height. Both variables are determined by other causes (chiefly genetics and childhood nutrition). Those underlying causes are the ones that need to be identified and manipulated in any causal system.

Some of the responsibility for this kind of misinterpretation 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.

Satisfying the third condition of a causal relationship requires investigating the causal system. Many good examples of causal analysis efforts in software engineering have been published [1, 2, 3, 4]. However, these efforts have adopted different terminology and approaches. In particular, the elements of a *causal system* have not been defined in a consistent way. The differences between the analysis procedures obscure the commonality in the subject matter to which the procedures are applied. Further complicating the situation are substantial differences in the

Figure 1: *Example of Correlation Between Variables*



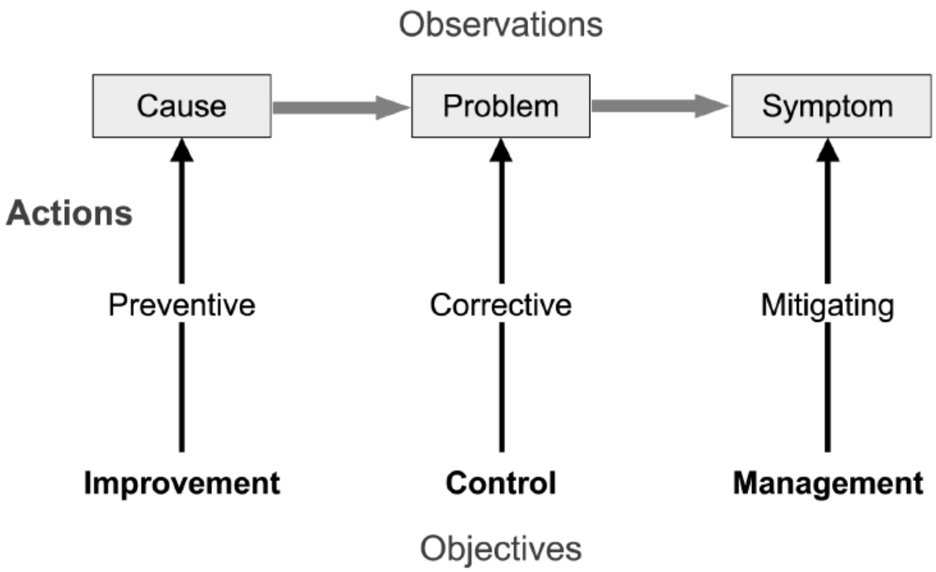


Figure 2: *Elements of a Causal System*

notion of causal analysis defined in the Capability Maturity Model© (CMM©) [5] and CMM Integration℠ [6] (described later).

One of the consequences of a poor understanding of the nature of causal systems and causal analysis is that causal analysis sessions become superficial exercises that do not 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. This article describes a model of causal analysis and a set of supporting terms that have evolved from extensive experience with the software industry. Some of these experiences with causal analysis were summarized in [7]. This experience encompasses scientific data-processing software, configuration management, and other software-related processes.

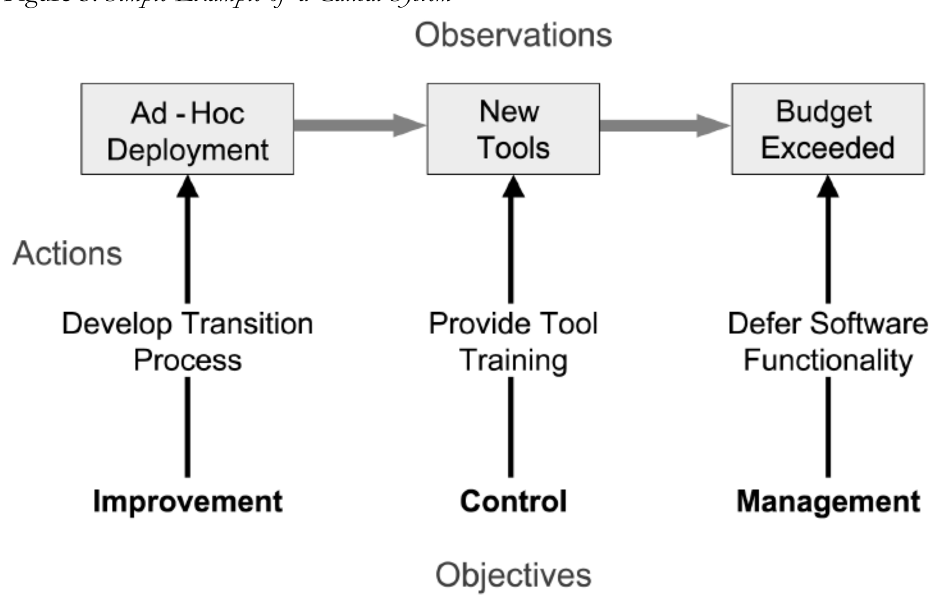
Elements of a Causal System

A cause-effect relationship may be one link in a potentially infinite network of *causes* and *effects*. A richer vocabulary than just causes and effects is needed to help us determine appropriate starting and stopping points for causal analysis. The model and terminology described in this section facilitate reasoning about causal systems and planning for causal analysis. Figure 2 describes the key elements of a causal system. Most of the approaches to causal analysis previously cited do not explicitly address all these elements of a causal system.

As indicated in the figure, causal systems include three classes of elements:

- **Objectives.** Our purposes in investigating the causal system.
- **Observations**. The events and conditions that comprise the causal system.
- **Actions.** Our efforts to influence the behavior of the causal system.

Figure 3: *Simple Example of a Causal System*



Observations are events and conditions that may be detected. Building an understanding of a causal system requires identifying these events and conditions, as well as discovering the relationships among them. Observations include the following:

- **Symptom**. These are undesirable consequences of the problem. Treating them does not make the problem go away, but may minimize the damage.
- **Problem.** This is the specific situation that, if corrected, results in the disappearance of further symptoms.
- **Cause**. These are the events and conditions that contribute to the occurrence of the problem. Addressing them helps prevent future similar problems.

Note that both problems and symptoms are effects of one or more underlying causes. Once a causal system is understood, action can be taken to change its behavior and/or impact on the organization. Actions may be of three types:

- **Preventive.** Reducing the chances that similar problems will occur again.
- **Corrective.** Fixing problems directly.
- **Mitigating.** Countering the adverse consequences (symptoms) of problems.

The corrective type usually includes actions to detect problems earlier so that they can be corrected before they produce symptoms. The optimum mix of preventive, corrective, and mitigating actions to be applied to a causal system depends on the cost of taking the actions as well as the magnitude of symptoms produced. Attacking the cause itself may not be the course of action that produces the maximum cost benefit in all situations. Potential symptoms and mitigations may be addressed as part of a risk-management activity.

Three objectives or motivations for undertaking causal analysis are common:

- **Improvement.** Triggered by recognition of an opportunity.
- **Control.** Triggered by an outlier or usual result relative to past performance.
- **Management.** Triggered by a departure from plans or targets.

Regardless of the motivation for causal analysis, all elements of the causal

® The Capability Maturity Model, CMM, and CMMI are registered in the U.S. Patent and Trademark Office by Carbegie Mellon University.
℠ CMM Integration os a service mark of Carnegie Mellon Univeristy.

system (as described earlier) should be considered.

Most real causal systems are more complex than Figure 2 suggests. That is, a specific problem may produce multiple symptoms. Moreover, many causes may contribute to the problem. Consequently, many different actions of all types may be possible. The Ishikawa diagram [8] is a popular tool for describing and reasoning about causal systems.

These general concepts of causal systems can be applied to the investigation of any undesirable situation, not just to the investigation of defects. Figure 3 shows an example of a causal system explaining a cost problem.

In the hypothetical example of Figure 3, a project has exceeded its budget for the work accomplished to date. This is a *symptom* of an underlying problem. It might be overcome through management action by reducing the functionality of the software to be delivered, thus reducing the remaining work.

A causal analysis of the situation might reveal that the adoption of a new tool suite without preparation by the project had reduced productivity. That is the *problem*. Providing training might increase the efficiency and effectiveness of the team, returning productivity to its normal state.

Preventing future occurrences of such problems might be accomplished by establishing a formal process for deploying new technology that assures appropriate training is provided. Whether or not such an action is taken to prevent this cause, correct the problem, or mitigate the symptoms depends on their costs and expected benefits.

For example, if the project has already passed through the phase where the tool suite was expected to have the greatest impact, then providing training to this project team may not be cost-effective, although preventing future occurrences and mitigating the impact of the current problem may still be helpful.

CMM/CMMI Views of Causal Analysis

While the software community's interest in causal analysis predates the publication of the CMM [5], the pursuit of process maturity has become a primary motivation for the adoption of causal analysis practices today. Both the CMM and CMMI [6] contain process areas describing causal analysis activities. These are Defect Prevention (DP) and Causal Analysis and Resolution (CAR), respectively. These two views of causal analysis differ in three important respects:

1. Required practices (activities).
2. Focus on prevention of defects.
3. Scope of triggering anomaly.

These differences are summarized below. The principal activities of the DP key process area of the CMM [5] are as follows:

- A DP plan is developed.
- Task kick-off meetings are held.
- Causal analysis meetings are held.
- Teams meet to coordinate actions.
- Defect prevention data is documented.
- Organizational process is revised.
- Project process is revised.
- Feedback is provided to staff.

Using the terminology described earlier, DP views *defects* as problems to be corrected. Failures are the consequences or symptoms of these problems that may have to be mitigated with workarounds, etc. The conditions that lead to the creation of defects are the causes to be prevented.

> *"One of the consequences of a poor understanding of the nature of causal systems and causal analysis is that causal analysis sessions become superficial exercises that do not look deeply enough to find the important causes and potential actions that offer real leverage in changing performance."*

The specific practices of the CAR process area of CMMI [6] are as follows:

- Select defect data for analysis.
- Analyze causes.
- Implement the action proposal.
- Evaluate the effect of changes.
- Record data.

Note that DP requires some additional activities, not obvious in CAR. In particular, developing a DP plan and conducting task (usually phase) kick-off meetings exceed the explicit requirements of CAR. DP is triggered by the recognition of an opportunity for improvement, e.g., a large number of defects associated with a particular activity.

DP focuses on developing preventive actions, rather than corrective or mitigating actions. Moreover, DP focuses on defects and their causes, not problems in general. Actions to detect defects earlier usually are considered *preventive* actions, although the case can be made that they really are corrective actions.

CAR is more general than DP. CAR *defines* a defect to include a broad range of problems. Any anomaly, outlier, or opportunity (as described in the preceding section) may trigger CAR, and result in any of the three types of actions identified earlier. It does not focus on prevention and early detection.

The generality of CAR makes it easy to come up with an example of investigating something to identify some kind of action, and thus claim satisfaction of the CAR requirements. On the other hand, systematic causal analysis does not need to be limited to the prevention of defects, as implied by DP. An understanding of the nature of causal systems helps to overcome the generality of CAR and ensure that each potential trigger for causal analysis is handled appropriately.

Summary

A good understanding of the basic concepts and terminology of causal systems helps to overcome the difficulties inherent in implementing a practice that seems *obvious*. The differences between the perspectives of CAR and DP has led to some problems as organizations either 1) try to facilitate the transition to CMMI by building a CMM Level 5 process that incorporates CMMI guidance into its initial design, or 2) transition an established CMM Level 5 organization to CMMI.

A causal analysis process based on CAR usually does not satisfy the CMM requirements for DP. A causal analysis process based on DP usually does not satisfy CMMI requirements for CAR. Understanding and applying the basic concepts of causal analysis underlying both process areas makes it possible to design a process that satisfies both sets of requirements.

Effective causal analysis is becoming even more important to the software industry as process maturity increases and new forces, such as Six Sigma [9] focus increasing attention on quality improvement. Academic researchers, especially those conducting empirical studies, also may benefit from thinking a little more systematically about causal systems. Application of the concepts and terminology presented here helps ensure that causal systems get fully investigated and effective actions are taken.◆

References

1. Mays, R., et al. “Experiences With Defect Prevention.” IBM Systems Journal Jan. 1990.
2. Dangerfield, O., et al. “Defect Causal Analysis – A Report From the Field.” ASQC International Conference on Software Quality, Oct. 1992.
3. Yu, W. “A Software Fault Prevention Approach in Coding and Root Cause Analysis.” Bell Labs Technical Journal Apr. 1998.
4. Leszak, M., et al. “Classification and Evaluation of Defects in a Project Perspective.” Journal of Systems and Software Apr. 2002.
5. Paulk, Mark, et al. Capability Maturity Model. Addison-Wesley, 1994.
6. CMMI Development Team. Capability Maturity Model – Integration, Vers. 1.1. Pittsburgh, PA: Software Engineering Institute, 2001.
7. Card, D. “Learning From Our Mistakes With Defect Causal Analysis.” IEEE Software Jan. 1998.
8. Ishikawa, K. Guide to Quality Control. Asian Productivity Organization Press, 1986.
9. Harry, M., and R. Schroeder. Six Sigma. Doubleday, 2000.

About the Author

David N. Card is a fellow of the Software Productivity Consortium, where he provides technical leadership in software measurement and process improvement. During 15 years at Computer Sciences Corporation, he spent six years as the director of Software Process and Measurement, one year as a resident affiliate at the Software Engineering Institute, and seven years as a member or manager of the research team supporting the NASA Software Engineering Laboratory. Card is the editor-in-chief of the *Journal of Systems and Software*. He is the author of “Measuring Software Design Quality,” co-author of “Practical Software Measurement,” and co-editor of “ISO/IEC standard 15939:2002 Software Measurement Process.” Card is a senior member of the American Society for Quality.

Software Productivity Consortium
2214 Rock Hill RD
Herndon, VA 20170
Phone: (703) 742-7199
Fax: (703) 742-7200
E-mail: card@software.org