ODC - a 10x for Root Cause Analysis

Ram Chillarege
Chillarege Inc.
April 2006
ram@chillarege.com, +1 (917) 790 9390, www.chillarege.com

***Abstract* -- Orthogonal Defect Classification (ODC) allows us to do a "10x" on Root Cause Analysis (RCA). It is a 10x in terms of the time it takes to perform root cause analysis and a 10x in terms of the coverage on the defect stream. These productivity enhancements are achieved by raising the level of abstraction and systematizing the analysis methodology. The impact of this productivity boost is far-reaching in its business impact with reported gains that are enormous.**

Introduction

Root cause analysis is the staple of productivity improvement methods in software engineering. Its success is legendary but the degree of implementation is often shallow. The difficulties in implementation are tied to cost, coverage, and skills needed to execute an effective root cause analysis. This puts an enormous strain on the manager who wants to use root cause analysis to understand issues and take action, but is unable to do so with speed and simplicity. As a consequence many opportunities which exist for quality improvement remain unexplored and unexploited.

The benefits of root cause analysis are usually quite obvious but the work load it generates is not as obvious. As a consequence it is not surprising to see institutions that begin such an exercise with gusto only to see the efforts diminish with time. Thus benefits are reaped only initially and the methodology and practice gradually fade from the organization.

ODC changes the game of how root cause analysis is done. It shifts the workload from detailed analysis of individual defects to a rapid categorization process that extracts the semantics from defects without detailed defect root cause analysis. Groups of defects are then analyzed using a measurement model that relates cause-and-effect. The skills used for this analysis are concentrated among few analysts as opposed to requiring every engineer to become an expert in root cause analysis. The quantification process through ODC helps prioritize issues so that we achieve the maximum impact for a given opportunity.

Classical RCA

Root cause analysis using defects can be performed in different ways, and there is no one preferred method. Most analyses are qualitative in nature and free-form. The classical root cause analysis method involves a subject matter expert studying a defect incident in considerable detail after the event is resolved. The objective is to yield a "root cause" which can be eliminated through specific actions.

This requires every individual performing the root cause analysis to have the skills and training to do so. This could often include every engineer given the sheer volume of defects in software development. Additionally, the subject matter expertise needed for such analysis is often distributed across the organization.

The first problem is that due to volume. If each defect takes an hour to five hours to identify the root cause, an organization with hundreds of defects will spend a significant fraction of its engineering time on root cause analysis.

The second problem is that due to the level of abstraction. Individual defects occur because of numerous factors: people, process, tools, legacy, complexity, and environment. To map the individual defect phenomena to this high level of abstraction is hard when we are looking at defects individually. We need to look at collections of them and be able to assimilate cause at the right process level.

Finally there is the big question of whether such reflection and analysis can be done by every engineer. It is certainly a skill that needs to be trained and one that does not necessarily fit every individual. An analogy is the contrast between individual contributors and managers. While we may have several bright people who perform well as individual contributors, not all make good managers. The task of finding root cause at a process level which is actionable; balancing competing factors requires experience and understanding far beyond the information captured within any one defect.

All these factors finally resolve to a cost equation that poses the hurdle. Typical analysis cost runs around 1 person hour per defect. Action implementation and follow-up ride on top of this base. Thus, even a small product release, which can have 1000 defects, has a burden rate that becomes the nemesis of root cause analysis. At IBM, in little over a year following the corporate imperative of the Defect Prevention Process (circa 1991) the practice thinned out and almost vanished. It was not merely the resource and cost, since return on investment was well established, but the fatigue and qualitative nature, which became hard to scale and maintain.

ODC

ODC is a technology that uses the defect data as an information source on the development process and the product. It is built on the recognition that individual defects capture a lot of semantic information about the development process and the product. ODC extracts the semantic information from the defect and turns it into a measurement on the process. The different attributes in ODC measure different aspects of the development process.

For instance, the Defect Type measures the advancement of the product through the development process. The Trigger measures the nature of testing being conducted within a particular phase. The combination of the Defect Type and Trigger, measured as a cross product, (contingency table since it is categorical data), gives us a measure of the effectiveness of the testing for a particular maturity of product development. The Defect Type and Trigger categories have been carefully designed so that they provide a measurement in their distribution for the aspect they capture. The Impact together with the Severity measures the nature of pain in the degree of pain that defects would cause a customer. The attributes Source and Age capture the origin of the code and the generation of legacy. Collectively these attributes captured an enormous amount of information from individual defects. And when a collection of defects are studied these data help develop considerable insight into the product and the process.

Figure 1 shows these attributes drawn around the circle that represents a defect. The two attributes on the left side of the circle are associated with cause. The "cause" in the process sense is associated with development and test. The attributes on the right side of the defect are associated with effect. The "effect" is with respect to the customer. Each defect thus creates a link between aspects

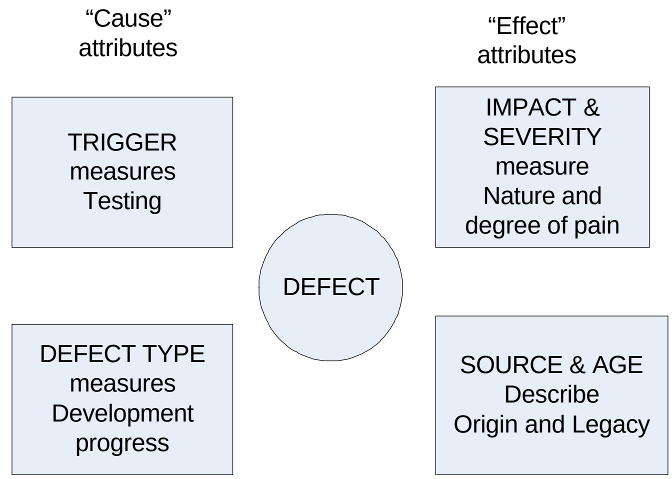

Figure 1. ODC attributes extracted from a defect. The Impact, Severity, and Trigger when the defect is found; Defect Type, Source, and Age when the defect is fixed.

that are associated with cause to those with effect. When we look at a group of defects, these data build a model that relates cause to effect.

The ODC based root cause analysis exploits this relationship to build models and perform analysis rapidly. The multi-dimensional data lets us explore factors that could be captured along with these core ODC measurements. Selective slices of the data allow for drill down and can be combined with classical root cause analysis on smaller subsets of data.

I like to describe ODC as the equivalent of an "MRI" for the software development process. It gives us the ability to picture the product advancing through the development process by analyzing the defect stream that is produced as a by-product of development. Thus, the analogy with the medical MRI technology.

ODC – RCA

ODC changes this equation for root cause analysis making it far more practical and scalable. The cost of analysis is reduced from 1 hour per defect to around 4 minutes per defect.

After 8 hours of training students classify defects at less than 4 minutes per defect. In a couple weeks, with continued practice, many achieve a classification speed of 2 minutes per defect, in retrospective mode. The in-process classification costs are negligible since it is dwarfed by the overhead of defect tracking. At this low cost, all the defects in a process can be classified and are subject to analysis – as compared to the classical root cause analysis, which is usually limited to a sample.

Classifying defects by ODC is only the first step. The actual root cause analysis is done by an ODC process analyst with skills in multi-dimensional analysis and statistics. Exploratory analysis usually guides a more detailed trend analysis and relationship understanding. The role of analysis is relegated to a set of experts who have a larger view of trends and organizational issues. The results of the analysis are shared with the team who relate the identified trends with potential solutions.

This quantitative approach to root cause analysis has multiple side benefits:

1. Not everyone in the development team needs to be involved in the root cause analysis.
2. There is greater coverage of the defect data given lower execution costs.
3. The quantitative methods allow easier comparisons from one release to another.
4. When multiple actions are involved, the data makes it far more tractable to prioritize and roll up actions.
5. Finally, communicating the results is far more systematic.

These benefits have very significant large system implications. They allow the methods to be scaled to larger projects and rolled out to organizations more readily, yielding a larger impact. Two case studies are noteworthy.

A large IBM product needed to go through a multi-year quality improvement to reduce cost of operations. The goal was to significantly enhance the code quality and reduce maintenance costs. After a couple years of classical methods of quality improvement, a 4x improvement in quality was accomplished but the improvement then plateaued. Over the following next few years ODC driven analysis and feedback yielded a ~15x improvement in quality. The original starting point was in the range of ~1500 defects per million lines of code – and after ODC it reached ~20 defects per million lines of code, resulting in code quality that rivals the best in the industry. The overall savings were ~$100 Million. Since warranty costs accrue annually, the total savings over the life of product is even higher.

A Nortel implementation of ODC for root cause analysis and directing development to strategically tackle a difficult development situation has a similar story. In about 5 years, the overall savings exceeded $250 million, when including reduced cost in warranty, critical situation handling and critical accounts.

These two case studies, with huge dollar savings reflect the more sophisticated implementations of ODC. They include not only the classification of data by ODC, but in addition a system of analysis methods, organization models, and customer profiles with an apparatus to deliver them to line management for execution.

Summary

ODC can significantly alter the economics and viability of root cause analysis by reducing the time it takes to perform the work and at the same time allow for greater coverage of the defect space. As a practical matter, it removes the critical hurdles that block the widespread use or root cause analysis, particularly when the defect volumes are large and the skills of the engineering team are limited. This factor helps the long term institutionalization on of this best-practice, providing not only short term savings but also long term strategic advantage.

References

[1] *"Orthogonal Defect Classification - A Concept for In-Process Measurements*", Ram Chillarege, Inderpal S. Bhandari, Jarir K. Chaar, Michael J. Halliday, Diane S. Moebus, Bonnie K. Ray, Man-Yuen Wong, IEEE Transactions on Software Engineering, Vol 18, No. 11, Nov 1992.

[2] *"Test and Development Process Retrospective - A case study using ODC Triggers"*, Ram Chillarege and Kothanda Ram Prasad, Proceedings International Conference on Dependable Systems and Networks, IEEE, 2002.

[3] *"Improving Software Testing using ODC - Three Case Studies"*, M. Buther, M. Murino, T. Kratcher, IBM Systems Journal, Vol 41, No. 1, 2002.

[4] "*Reflections on Industry Trends and Experimental Research in Dependability*", Daniel Siewiorek, Ram Chillarege, Zbigniew Kalbarczyk, IEEE Transactions on Dependable and Secure Computing, Vol No.2, 2004.

[5] "*Rapid Business Value Delivery through Software Engineering*", Dr. Rama Munikoti, Keynote, 10th Intl. Symposium on Software Reliability Engineering, IEEE, Boca Raton, FL, Nov 4, 1999.

Author

Ram Chillarege is the inventor of Orthogonal Defect Classification. He has a consulting firm that specializes in software engineering optimization, and has a range of training services in ODC. Prior to starting his consulting firm, he was executive vice president of Software at Opus360. He was at IBM for 14 years, where he founded and headed IBM's center for software engineering. He was also the key driving force to set up IBM's corporate wide software testing center of excellence. He is an IEEE Fellow, and author of ~50 technical articles. He earned a Ph.D. in computer engineering from the University of Illinois, Urbana-Champaign, B.E. and M.E. from the Indian Institute of Science, and B.Sc. from University of Mysore.