Monitoring and Improving the Quality of ODC Data using the "ODC Harmony Matrices": A Case Study

Nirav Saraiya and Jason E. Lohner
Motorola Inc.
{nsaraiy1, jlohner1}@motorola.com

Jongmoon Baik
Information and Communications University
jbaik@icu.ac.kr

Abstract

Orthogonal Defect Classification (ODC) is an advanced software engineering technique to provide in-process feedback to developers and testers using defect data. ODC institutionalization in a large organization involves some challenging roadblocks such as the poor quality of the collected data leading to wrong analysis. In this paper, we have proposed a technique ('Harmony Matrix') to improve the data collection process. The ODC Harmony Matrix has useful applications. At the individual defect level, results can be used to raise alerts to practitioners at the point of data collection if a low probability combination is chosen. At the higher level, the ODC Harmony Matrix helps in monitoring the quality of the collected ODC data. The ODC Harmony Matrix complements other approaches to monitor and enhances the ODC data collection process and helps in successful ODC institutionalization, ultimately improving both the product and the process. The paper also describes precautions to take while using this approach.

1. Introduction

Orthogonal Defect Classification (ODC) is a measurement system for software processes based on the semantic information contained in the defect stream [3]. ODC is a means of capturing actual defect data in real-time and providing "in-process" feedback to the developers, testers, and management. ODC was invented at IBM in the 1990s. Since its conception, IBM has continuously updated the ODC to address all relevant and contemporary issues of software product development. Today, ODC is being used by many other organizations including Motorola. In this paper we have not explained the details of ODC. Interested readers can obtain information directly from IBM [5].

While it is very easy to understand and discuss ODC conceptually, its actual implementation involves some challenges. ODC practitioners often find the defect classification ambiguous and often challenge the "orthogonality" of the attribute set [4]. To institutionalize ODC, it is certainly important to educate and involve top management but it is even more important to train development and test engineers (i.e. ODC practitioners) about actual defect classification. We have found that if ODC practitioners are not well trained, the end result is a collection of incorrect defect classification data that leads to meaningless results. This baffles upper management as they are not able to see the promised results of ODC. At the same time, when the practitioners do not see any meaningful actions resulting from the collected ODC data, so they are further dissuaded from using it. This creates a huge gap between management and practitioners. Thus, in our opinion, general education about ODC in theory is important but not a critical task. The critical task is to collect meaningful and correct defect classification data.

The element of time is also an important factor that needs to be addressed. We observed that many a time when release ODC signature data is analyzed, the validity of the data is questioned. At that time, the release is well past its development phase and the development organization is then involved with the next release. Thus, there is no motivation for developers to go back and correct the ODC data. The result is that decision-makers are left with incorrect data and it is then too late to do anything about it. This results into lost effort of data collection and an increased gap between wish-list and reality.

In this paper we describe the process of developing the 'ODC Harmony Matrix', which is all about finding meaningful relationships between Type-Trigger combinations. Using the meaningful relationship set, we propose some interesting applications of these

Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)
0-7695-2656-X/06 $20.00 © 2006 IEEE

criteria to improve the quality of ODC data which ultimately helps both the product and process.

2. Define Phase and Survey Other Approaches

2.1. Define:

While searching for ways to address the problems of properly classifying ODC data mentioned in the previous section, we found that one means of addressing the issue is to provide hands-on training to practitioners by presenting the ‘Line of Attack Algorithms’ [4]. We also found that developing a quantitative baseline is another approach that supplements the “Line of Attack” Algorithms. [6] Thus, an “in-process” tool exists to monitor the progress of a development project and the quality of the collected data.

Both of the aforementioned approaches solve a specific aspect of the problem and are complementary in nature. We decided to divulge further and defined our problem statement as: “Need to explore any co-relation between defect type and development trigger combinations that could help ODC practitioners to select the correct combination and assist in monitoring the quality of collected ODC data.”

2.2. Measure:

ODC has been in use in one at of the Motorola organizations for many years. The organization also has an Escaped Defect Analysis (EDA) team that verifies ODC data while carrying out EDA analysis. The EDA team consists of experienced engineers from different functional areas with solid ODC experience. We used the ODC data verified by the EDA team as our data set.

To begin, we applied our understanding of the development process and used our experience to comprise a matrix of defect type versus development trigger. We primarily classified each combination in terms of three levels of probabilities of occurrence represented as Low, Medium, and High. Our initial matrix represented our hypothesis. After formulating our hypothesis, we extracted ODC data from the EDA database and plotted a matrix of Type versus Trigger.

We observed a correlation among Type-Trigger combinations and our initial hypothesis was correct to a great extent. We had limited data for some of the combinations making it difficult to verify everything, but we found that the data did indeed provided substantial support to our hypothesis.

In the meantime, we also consulted experienced members of the EDA team (i.e. domain experts) and revised our results by incorporating their suggestions. The final result, representing our initial proposal, was a defect type versus defect trigger matrix representing the probability of occurrence of different combinations in terms of the three distinct levels. The resultant matrix as shown in Table 1 is supported by the trends of EDA data for most cases while over-ridden by domain experts’ opinions for a few of the combinations.

Table 1: Initial ODC Harmony Matrix

Trigger	Develoment Type							
	ALG	ASN	BLD	CHK	DOC	FUN	INT	TIM
Backward Compatibility	H	M	H	H	L	L	L	L
Concurrency	M	M	L	M	L	L	L	H
Doc. Consistency/Standards	L	L	L	L	H	L	L	L
Language Dependencies	L	H	L	H	L	L	L	L
Lateral Compatibility	M	H	L	M	L	M	H	H
Previous Phase	H	M	L	H	L	M	M	L
Rare Situation	M	M	L	H	L	L	L	H
Side Effects	M	M	L	H	L	L	L	H
Understanding Flow	H	M	L	H	L	L	M	L

We quickly realized that this initial ODC Harmony Matrix may not provide substantial benefit to the practicing organization. We found through practical experience that it is common for an engineer classifying their faults, or for practitioners, to clearly identify either the defect type or the trigger during analysis. Therefore, aligning with our goal we wanted to assist them in correctly completing the classification. We created a Harmony Matrix normalized by defect type and another Harmony Matrix normalized by defect trigger. These two distinct matrices can be found in the Results section of this document. When the confidence of one of the categories is high, the appropriate matrix will guide practitioners to a probable combination for evaluation.

2.3. Analysis:

In this section a detailed discussion of the rationale behind our categorization is presented. The intent of this section is to provide details behind our reasoning.

Prior to using historical data it was required to ensure the integrity of the data. Therefore, practitioners performed a scrub of a percentage of the EDA data set we intended to use. The goal of the scrub was to discuss the validity of the data and make all necessary corrections by following the “Line of Attack” Algorithms to institutionalize ODC [4]. The resultant

Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)
0-7695-2656-X/06 $20.00 © 2006 IEEE

data set was used as the premise for our Normalized Harmony Matrices.

The categorization into levels was given a great deal of consideration. The Harmony Matrix is intended to assist practitioners by providing confidence ratings referred to as probabilities. We have identified the three probability categories of Low, Medium, and High.

The category of Low actually represents both a low probability to practically no probability as it is inherent that there are a few combinations that do not make practical sense. High as a category was fairly easy to determine. The numbers clearly identified a large number of the high-probability cells. Cases exist where a cell has been identified as high probability due to the fact that it is a cell that we want to be considered before moving on to other candidate cells in order to complement existing ODC classification approaches. The Medium category is a grey area indicating that the combination is noticed to an extent and that it is worth giving appropriate consideration after ruling out the high probability cells.

While creating the normalized matrices it was easier to correlate the data into the appropriate categories. Figure 1 represents how clear the high categories were defined by the EDA data as indicated by the data points on the graph.

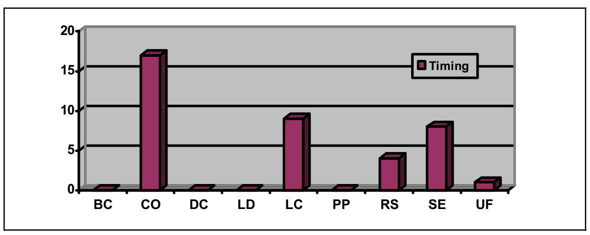

Figure 1: Normalization Example

Error! Reference source not found.2 represents an example of how we translated the data points into the probability categories. With respect to the timing defect type, we made a conscious decision to classify LC as high and SE as medium due to the nature of the line of attack algorithm even though they had similar probability according to the historical data. Our decision was made to align with and complement the line of attack algorithm where LC is second from the top and SE second from the bottom. In the grand scope of the entire matrix it should be noted that similar conscious decisions were made in the awareness of existing, complementary approaches and this merely represents one example.

Table 2: Normalization Example – Categorizing

Trigger	Timing Type	
	Percent	Category
BC	0%	L
CO	44%	H
DC	0%	L
LD	0%	L
LC	23%	H
PP	0%	L
RS	10%	M
SE	20%	M
UF	3%	L
Totals	100%	

3. Use of the Harmony Matrices

In this section various uses of the ODC Harmony Matrices are presented and clearly outlined.

3.1. Practitioners:

The Harmony Matrices provide a very useful tool for practitioners to use in the process of classifying defect data. We observed that often times a practitioner is more confident about their selection of either defect type or development trigger but uncertain about the classification of the other. The appropriate Normalized Harmony Matrix is helpful in such situations by suggesting the attributes with higher probability of one category based on the committed selection of the other category. The Harmony Matrices also assists in leading practitioners to the best classification by eliminating any of the low-probability combinations.

In the case where the practitioner is confident about their classification of both defect type and development trigger without use of a matrix, the practitioner can verify his/her data against the Harmony Matrix normalized by Type as it is often the case that the defect type is clearly understood by the one performing the classification. If the combination lands into a low-probability zone, the practitioner is encouraged to review their choice. It should be noted that one should not change their selection just because it lands into a low probability zone. However, proper understanding of why the certain selection was made greatly helps one to understand their defect better.

This is one of the most potent uses of the Harmony Matrix. It provides instant “in-process” feedback to the practitioner about their selection and gives an opportunity for revision to any probable misclassifications. This greatly improves both the quality and reliability of the collected ODC data.

3.2. Development/Program Managers:

The Harmony Matrix normalized by Type is also useful to development and program managers. The

Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)
0-7695-2656-X/06 $20.00 © 2006 IEEE

progress of a development project can be monitored thru all phase of the DLC and appropriate action can be taken if many low-probability combinations are encountered. The classification of low-probability combinations can be investigated to determine if it is due to misclassification by development engineers. In the case of confirmed misclassification, the identification of ODC training needs is clear. However, it could also be a perfectly legitimate classification that simply required an explanation. In any case, understanding of why it was classified in low-probability zones always provides more insight into their process and product.

The most important advantage of the Harmony Matrices is that in case of any misclassifications it provides an opportunity to scrub ODC data “in-process”. The scrub of the data is completed while the engineer’s memory is still fresh and can recall the details of the problem. It is ideal that this would be prior to the team moving on to next task.

3.3. Growth Measurement:

The Harmony Matrix normalized by Type is also useful in comparing ODC expertise of different teams or of the same team over any given period of time. The number of low probability cells usually suggests misclassifications and hence lack of ODC usage expertise. As any team gains experience the number of initial classifications of low probability cells should decrease.

4. Results: Normalized Matrices by Type and Trigger

Hard work harnessing historical data and experience led to the creation of the Harmony Matrices. Each matrix represents normalization of ODC data, one by defect type and the other by trigger. Corresponding to each matrix is a set of data points used to correlate the categories plotted on the respective matrices.

When normalizing the data by the defect type, the matrix was built on the basis that the confidence is assumed to be high that the Type has been correctly identified. By scanning the corresponding column for the selected defect type, the probabilities are indicated of what one can expect the corresponding Trigger to be. With respect to the Trigger, the same holds true except that one would scan the corresponding row for the selected Trigger.

The following Harmony Matrix has been normalized by Type. It is the case that practitioners are mostly certain of classification for defect type. Therefore, this matrix represents the one that can be used as a check against classifications “in-process”, for monitoring the progress of a development project, or as the growth measurement of a team.

Table 3: Harmony Matrix Normalized by Type

Defect Trigger	Development Defect Type							
	ALG	ASN	BLD	CHK	DOC	FUN	INT	TIM
Backward Compatibility	H	M	H	M	L	M	L	L
Concurrency	M	M	L	M	L	M	L	H
Doc. Consistency/Standards	L	L	L	L	H	L	L	L
Language Dependencies	L	M	L	L	L	L	L	L
Lateral Compatibility	M	H	L	M	L	H	H	H
Previous Phase	H	M	L	M	L	M	M	L
Rare Situation	L	L	L	L	L	L	L	M
Side Effects	L	L	L	L	L	L	L	M
Understanding Flow	H	H	L	H	L	L	L	L

Table 4: Probability by Trigger and Type

Trigger	Development Type								Totals
	ALG	ASN	BLD	CHK	DOC	FUN	INT	TIM	
BC	8%	5%	20%	13%	0%	12%	3%	0%	8%
CO	6%	3%	20%	9%	0%	0%	3%	44%	10%
DC	3%	2%	0%	2%	67%	12%	6%	0%	4%
LD	0%	3%	0%	0%	0%	0%	0%	0%	1%
LC	24%	33%	20%	9%	0%	48%	53%	23%	26%
PP	12%	21%	0%	10%	0%	12%	6%	0%	11%
RS	5%	5%	20%	10%	0%	0%	9%	10%	7%
SE	3%	2%	0%	4%	0%	4%	6%	21%	5%
UF	40%	26%	20%	44%	33%	12%	15%	3%	29%
Totals	100%	100%	100%	100%	100%	100%	100%	100%	100%

The following Harmony Matrix has been normalized by Trigger. This matrix is used to assist practitioners in the instance that they are unable to classify the defect type and have confidence in what the Trigger is.

Table 5: Harmony Matrix Normalized by Trigger

Defect Trigger	Development Defect Type							
	ALG	ASN	BLD	CHK	DOC	FUN	INT	TIM
Backward Compatibility	H	M	H	H	L	L	L	L
Concurrency	M	M	L	M	L	L	L	H
Doc. Consistency/Standards	L	L	L	L	H	L	L	L
Language Dependencies	L	H	L	H	L	L	L	L
Lateral Compatibility	M	H	L	M	L	M	H	H
Previous Phase	H	M	L	H	L	M	M	L
Rare Situation	M	M	L	H	L	L	L	H
Side Effects	M	M	L	H	L	L	L	H
Understanding Flow	H	M	L	H	L	L	M	L

Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)
0-7695-2656-X/06 $20.00 © 2006 IEEE

Table 6: Percentage of Type – Trigger Known

Trigger	Development Type								Totals
	ALG	ASN	BLD	CHK	DOC	FUN	INT	TIM	
BC	24%	12%	4%	44%	0%	12%	4%	0%	100%
CO	15%	6%	3%	21%	0%	0%	3%	52%	100%
DC	17%	8%	0%	17%	17%	25%	17%	0%	100%
LD	0%	100%	0%	0%	0%	0%	0%	0%	100%
LC	22%	23%	1%	8%	0%	14%	21%	10%	100%
PP	26%	37%	0%	23%	0%	9%	6%	0%	100%
RS	17%	13%	4%	35%	0%	0%	13%	17%	100%
SE	12%	6%	0%	18%	0%	6%	12%	47%	100%
UF	33%	17%	1%	38%	1%	3%	5%	1%	100%
Totals	24%	19%	2%	25%	1%	8%	10%	12%	100%

5. Conclusion and Recommendations

ODC offers potential benefits by gathering critical information about defects throughout the development life cycle. However, the potential is unable to be harnessed if ODC is not institutionalized properly [4]. It is therefore crucial that the integrity of the data not be compromised and that practitioners accurately classify the defect type and development trigger. While ODC institutionalization continues to face challenges, we have identified the ODC Harmony Matrices as a part of the solution.

The Harmony Matrices are a result of our ODC experience. We have examined historical data for trends and provided guidelines of what the generally acceptable combinations of defect type and development trigger are within our organization. The guidelines are presented in a matrix and can be used to assist in determining the probable combination considering there is confidence in what either the defect type or development trigger is.

These Harmony Matrices do not suggest that the classification is mechanical. The ODC classification is still a human-intensive task and is not automated by adopting this matrix. The Harmony Matrices help in the subjective process by making the defect classification less objective [4].

We are confident that use of these matrices results in a more consistent and accurate data collection, an "in-process" scrub of ODC data, and better institutionalization of ODC in our organization.

It should be pointed out that the Harmony Matrices represent a suggestion based on historical data and the logical relationships among different Types and Triggers. Simply because some defect classifications fall into the low-probability zone does not mean that a practitioner should reject that combination. Also, simply because the combination falls into the high-probability zone does not have a one-to-one correlation that it is a correct classification, because for any given Type-Trigger combination it is probable that more than one high-probability combination is possible. Thus, the use of the Harmony Matrices is not a replacement for thinking. Common sense and application shall always prevail and remain on top of all norms.

We also recommend that an organization should have a mechanism in place to monitor the effectiveness of the categories defined in the Harmony Matrices and modify the matrix accordingly with reasonable data to support the modification. This is important to ensure that the Harmony Matrices reflect the reality of their contemporary product and process.

References

[1] Ram Chillarege, "Orthogonal Defect Classification",
Chapter 9, Handbook of Software Reliability Engineering, Michael R. Lyu, Editor, McGraw-Hill 1996

[2] Ram Chillarege, et. al., "Orthogonal Defect Classification – A Concept for In-Process Measurements", IEEE Transactions on Software Engineering, Vol. 18, No. 11, Nov 1992.

[3] IBM Research, Center for Software Engineering, "Details of ODC v 5.11", http://www.research.ibm.com/softeng/ODC/DETODC.HTM.

[4] Nirav Saraiya, Cvetan Redzic, Jongmoon Baik, "Using 'Line of Attack' algorithms to institutionalize ODC", SERA 2004, Los Angeles

[5] IBM Research, Center for Software Engineering, "Orthogonal Defect Classification", http://www.research.ibm.com/softeng/ODC/ODC.HTM

[6] Nirav Saraiya, Premal Mehta, Cvetan Redzic, "Quantifying ODC defect type signature baseline: A case study", Motorola S3S 2004, Itasca