Abstract

In this paper, we present a process for bringing the organizational training department of a software development organization closer to the development projects. We envision an organizational training department that can proactively help the development projects in reducing their defect rate.
To achieve this goal we take advantage of the available data at CMMI maturity level 3 organizations. We show that by taking advantage of an established defect database system, the training department can cost-effectively implement an evaluation process that gives results at the Kirkpatrick’s “Results” level.

1. Introduction

Organizations nowadays invest large amounts of money in their training programs. Just in the United States, organizations invest an average of U$S 100 billion annually [1, 2]. Organizations invest in training under the assumption that higher trained employees will result in higher quality products and reduced costs.

Nevertheless, in spite of all this investment and effort, organizations have not been so successful at providing consistent data that will link the investment on Organizational Training to organizational results. We believe training managers will have to start showing results to their respective organizations in order to keep the training founding flowing.

Kirkpatrick’s [3] four levels of training evaluation model has been successful in guiding organizations to report their training results since it was first published in 1959. And yet, until 2008 not many organizations have been successful at providing tangible results at Kirkpatrick’s fourth level[4] (Results).

In this paper, we propose a mechanism for linking Organizational Training to defect data. Such a mechanism will enable software organizations to obtain results at the “Result Level” within reasonable costs and therefore maximizing training return on investment. We will show that by using defect data as an input to Organizational Training, we can provide a measure of Return of Investment (ROI) by using defect data as input.

Despite the fact that the idea of linking defect data to ROI is not novel, most authors use defect reduction and the cost of correcting the defects as the input for ROI calculations. For instance at NDIA2005 Olson[5] justifies the effort investment in implementing an Early Defect Detection System by using Defect Data. Humphrey justified the effort of implementing Team Software Process (TSP) by means of defect reduction. These proposals are first of all biased by the need of the authors to justify each methodology. In addition to this, they ([5-7]) are focused on process improvement, and neglect training to annotations within their papers. Finally, they do not address the broader subject of how an Organizational Training department within a software organization should justify its existence. Furthermore, in this paper we will give an outline of a concrete process that will integrate these ideas into a simple and cost effective organizational training process that will aid in the overall reduction of defects.

2. Organizational Training within the CMMI model

In order to set expectations, we will base our discussion thinking in a training department that follows a CMMI [8] Organizational Training compliant process. CMMI stands for Capability Maturity Model Integration. And it is a process improvement model that is stewarded by the Software Engineering Institute of Carnegie Mellon University. The CMMI model is divided into 22 process areas, being Organizational Training (OT), one of them.

In short, within CMMI the Organizational Training department must elicit training needs and align them to organizational objectives (OT, SP 1.1, SP 1.2). Those needs must then be translated into specific training events that should be documented in a training tactical plan that should be followed and managed (SP 1.3, SP 1.4). Training must then be delivered to the respective audiences (SP 2.1) and finally the organizational training department must communicate training results to higher management and training stakeholders.

3. Overview of Kirkpatrick’s four levels of training

Despite the fact that the idea of linking defect data to ROI is not novel, most authors use defect reduction and the cost of correcting the defects as the input for ROI calculations. For instance at NDIA2005 Olson[5] justifies the effort investment in implementing an Early Defect Detection System by using Defect Data. Humphrey justified the effort of implementing Team Software Process (TSP) by means of defect reduction. These proposals are first of all biased by the need of the authors to justify each methodology. In addition to this, they ([5-7]) are focused on process improvement, and neglect training to annotations within their papers. Finally, they do not address the broader subject of how an Organizational Training department within a software organization should justify its existence. Furthermore, in this paper we will give an outline of a concrete process that will integrate these ideas into a simple and cost effective organizational training process that will aid in the overall reduction of defects.
Kirkpatrick’s four[3] levels provide a framework after which training organizations can measure up. Kirkpatrick’s model was proposed in 1959, and has been extensively reviewed in the literature, a simple search in scholar.google.com shields over 900 references that cite either the original articles or the published book [3]. This shows that despite being half a century old, this model is still active and a reference for training evaluation. Nevertheless, some authors have proposed enhancements of Kirkpatrick’s model (for instance see [9-11]), but we have chosen to stick with the four original levels. Mainly, because we believe that the model simplicity is what has made it stand through time. We expect our proposals to abide by this simplicity and as a result to make them readily available for other practitioners to implement.

As we have mentioned earlier, Kirkpatrick defines four levels by which an organization should evaluate its training programs. This section provides a brief overview of each of the levels.

- **Reaction**: Refers to obtaining a measure of how participants felt during the training activity. Evaluating reaction is usually simple and cost-effective; it is usually accomplished by asking participants to fill in polls to the instructor. Reaction sheets are important, because through them we can get tangible results about variables that might affect learning results (like motivation, adequacy of resources, etc).

- **Learning**: Learning aims at obtaining a measure on the advance of a trainee at a certain skill or knowledge as a result of the training he has received. Learning measures are usually obtained by applying pre and post tests during a training event. The difference in performance between the two can be linked to the learning that has occurred in the training event. The importance of measuring learning resides in the assumption that no change in behavior can take place if no learning has occurred.

- **Behavior or Transfer**: Refers to the ability to obtain measures regarding how much of the acquired knowledge was successfully applied on the job. Transfer measures are usually harder and more expensive to obtain. A typical method would be to administer tests at a given point in time. Transfer measures, are the first level of measures that the organization should have interest in because those levels begin to show how the effort incurred in training has a return in the job.

- **Results**: The Results level is about bottom line results. It implies obtaining data about the degree in which training was able to influence in the organization objectives. Results data is about linking the impact of training in organizational results. An interesting point that Kirkpatrick notes is that isolating the training effects for a definite result measure is usually not cost-effective, since organizations take several actions in order to achieve their business goals. Trainers should look for “evidence beyond a reasonable doubt” that the training has had a positive influence on the bottom line. “Reasonable doubt” means that every organization starts a number of initiatives in order to achieve the desired results, but the training evaluation should be able to show by itself that it has positive impact in the organizations’ results.

Kirkpatrick’s book includes a couple of case studies of organization (Chapter 14 at Motorola University, Chapter 23 at Cisco) which have achieved measurement at several of the levels including examples of successfully implementing the four levels (Intel University in Chapter 18).

On the other hand, Alliger has shown that there have been few organizations that were able to show bottom line results. He has also tried to find correlations [4, 11] between the four levels, with the intention of trying to predict the results level without actually measuring it.

Our overall impression is that achieving measurement in all four levels is an effort intensive and costly task, and as a result we are driven to look for alternative paths in order to justify training investment. We would like to envisage an organizational training process that is able to give back metrics in Kirkpatrick’s four levels while making the climb through the levels as linear as possible.

### 4. Return on Investment in training

Return on Investment can be related to Kirkpatrick’s level 4. Even though Kirkpatrick specifically says that “level 4 measures should not be considered ROI”, we believe that ROI is a valid technique for assessing the benefits of a training program. Other authors like Phillips[9], have avoided contradicting Kirkpatrick by proposing an enhanced model with a level 5 which he has named Return on Investment. Nevertheless, we believe that it is not entirely necessary to make this distinction.

One of the most extensive works we have reviewed comes from Phillips[9], who acknowledges that calculating the ROI on training is a complex process, and he proposes a simple four step process to guide its calculation:

1. **Data Gathering stage**: Phillips’ ROI process is very much aligned to Kirkpatrick’s model. When data gathering is carried out data must be collected in all four of Kirkpatrick’s levels. Based on the metrics needs, the data gathering plan must develop the activities needed to collect, to store and to make the data available for use in the following steps of the ROI calculation process.
2. Isolate Variables that Can Affect Training: In order to effectively assess the ROI of training the variables that affect training must be isolated. This is especially important in level 4 evaluation where other variables than training can have influence in business objectives. In his book Phillips proposes several methods for isolating variables, being the most classical one the use of a control group. Training variables must be isolated so that the following levels can reach the conclusion that training has had bottom line effects “beyond reasonable doubt”.

3. Converting data to monetary Benefits: Phillips distinguishes between Hard and Soft data. Hard data: usually tangible units of outputs of a process, this is generally easy to gather since it’s usually the target process output. Soft data: the side effects of training programs that are not easily converted to monetary values (like employee turnover, job satisfaction and commitment, and absenteeism for example). Soft data must be included in the ROI calculation, since training is usually aimed at generating soft skills, and those variables can have influence in bottom line results. Usually, estimation and/or tabular scales must be implemented in order to include Soft data into ROI calculation.

4. Applying ROI classical formula for calculation: Finally, with all data at hand, a ROI calculation must be carried out.

**Equation 1 Classical ROI formula**

\[ ROI(\%) = \frac{\text{Benefits} - \text{Costs}}{\text{Costs}} \times 100 \]

When calculating Benefits of the training program, the variables to be considered must include both Hard and Soft data. Usually projected benefits include increased productivity. When estimating the projected benefits the most conservative estimation must be used. Costs must also include Hard and Soft data sources; training cost usually includes the cost of preparing and imparting the training, and the cost of having the trainees without producing in their regular tasks.

Phillips has been criticized in that he puts too much emphasis in Hard data [12]. In [13] there is a comprehensive review on the literature on Return of investment training initiatives. This review compares numerical approaches like Phillips [9] to more Human Resource Development approaches. Phillips is very general in his studies about ROI on training and his books are not specifically targeted at software. As we have mentioned, we found that Humphrey makes a brief calculation on the return of investment of deploying TSP within an organization in his book [14]. By using ROI calculation Humphrey justifies how the training effort of implementing pays off by taking into account the projected defect reduction.

Other ROI models come from Software Cost of Quality research. In [15] the authors present a model for the cost of quality that takes into account direct and indirect costs, then they move on into taking into account the projected savings of implementing the defect detection techniques.

Also Slaughter [16] establishes two techniques for evaluating the Return of Software quality. His techniques includes a Post Mortem analysis where he assigns the cost of reported defects.

Our work can be related to the ideas about ROI written in this book in that we too are aiming at linking defect reduction to training return on investment. But we differentiate in that we are proposing an integrated process that will take into account actual defect data from the project and use it as an input for the training process. ROI will be used to evaluate the overall effectiveness of implementing the process.

### 5. Orthogonal Defect Classification

In order to link training to defect reduction, a mechanism for managing defects is needed. At IBM [17, 18], Defect Data Classification has been used to drive Software Process Improvement initiatives, they call their process Orthogonal Defect Classification (ODC). Within the ODC context, developers classify defect data in Orthogonal classifications. IBM proposes a taxonomy for defect classification based on the source of the defects:

- **Communication**, in this category, the developer did not receive the required information or the information was incorrect.
- **Education**, in this category, the developer did not understand some aspect of the product or the process. This category is further divided into education in base code, education in new function, and other education types. depending on what was not understood.
- **Oversight**, in this category, the developer failed to consider all cases and conditions. Usually some detail of the problem or process was overlooked. The developer forgot, had difficulty checking thoroughly, or did not have enough time to be thorough.
- **Transcription**, in this category, the developer knew what to do and understood the work package in depth, but simply made a mistake. Transcription errors are typically caused by some problem in a procedure for example, typing or copying a list of items.

Classified defects are stored in a defect tracking system (what we will later reference as the QADatabase). Classified defects will start to populate this defect
tracking system, and will provide the data for Causal Analysis meetings.

Causal Analysis meetings are held at every product delivery, the aim of these meetings is to identify the sources of their defects. Within these meetings a sample of the defects is taken and solutions for removing the source of the defects are proposed.

IBM[17] calls these proposed solutions Action Items which are handled by an Action Team. The Action Team is then responsible for the selection of the actions which will be evaluated for process improvement, and for the implementation of the selected improvements. The Action Team will classify some of that action into process improvement actions and establish an action plan for the implementation of the improvements[18].

In short, this whole process is aimed at identifying opportunities for improvement in the current standard process.

Not only IBM but other organizations have implemented ODClike processes and have reported improvements in the form of reduced defects. Our review has shown [19] that most organizations tailor the defect classification schemes in order to suit their own process improvement needs. In [6] an overview of the defects classifications proposed is given. In every example, defects classifications were developed to suit each organization’s needs. For instance, despite the fact that HP classification scheme seems more exhaustive than IBM classification, HP and IBM report similar benefits.

6. A Proposal for an Organizational Training Process

We are envisioning a process that will tie ODC like processes to Organizational Training. We have shown that all the reviewed enterprises have used defect classification techniques as an input to software process improvement initiatives. We propose to also use defect data to drive training. Moreover, by linking organizational training to defect reduction, we will be able to quantify how training investment pays by giving results in the “Results” level by applying a ROI calculation. For illustration purposes, in Figure 1 we have laid out our process on top of the CMMI organizational framework.

As we have mentioned, the organization we are working with has recently had its process evaluated.

Figure 1 Overview of the proposed process
CMMI maturity level 3. This means the training department establishes the organization training needs based on the organizational business goals (arrows 1 and 2), which are represented by the Organizational Process Focus (OPF) process area.

Organizational Process Definition (OPD), will provide the process assets in which practitioners in the organization development projects must be trained (arrow 4). The training department will execute its processes in order to plan and provide the necessary training to the development projects (arrow 5). Depending on the implementation of the model, the training department can also assist development projects in satisfying particular training needs, as the CMMI model suggests in the Project Planning process area (PP SP 2.5, arrow 6).

In our proposal, when projects perform the Validation and Verification activities (arrow 7) they are forced to classify defects. Classified defects will populate the QADatabase (arrow 8). At every delivery, the project teams will carry out a causal analysis meeting. These meetings will take the classified defects (arrow 9) as input in order to identify defect sources and provide proposals for improvement (arrow 10). The training department will be proactively monitoring those proposals in order to design and implement training that will specifically address that project defect source (arrow 11).

Finally, defect reduction data can be used as input for a ROI calculation (arrow 12). We acknowledge that not all defects can be treated as process shortcomings, for our proposal we have only taken into consideration defects that have been classified as Education or Transcription failure. It is our impression that Oversight errors are part of the human nature, and such type of defects are more difficult to be addressed. Communication failures, on the other hand, can more clearly be an indication that the process is failing to supply the developers with the adequate channels of communication needed.

This creates a short feedback cycle within an ongoing project that will have positive impact in the project’s quality. Our proposed process will more rapidly return to the development projects the root cause analysis results than having to wait for a new process roll-out as ODC proposed.

7. Process Deployment Plan

We are at the early stages of the implementation of the full process. A constrain we must respect while implementing this process is that we want to make sure that we are inflicting as little overhead as possible onto the organization. As a result, we have sorted the activities of our process into three phases. The process deployment plan is shown in Table 1:

- a) Establish process capability (arrows 7 and 8 of the diagram). In this first phase, we want to make sure the organization has the necessary infrastructure to support the process. Activities for this phase include development and validation of a classification scheme, support tool modification, and analysis of current data.

- b) Deploy a Causal Analysis process (arrows 9 and 10 of the diagram). The deployment of a Causal Analysis process will require the development of such a process for the organization. Then the practitioners must be trained in the execution of the Causal Analysis process. In addition to this, the Causal Analysis meetings must be included as planned activities within the organization’s projects plans. Finally, a verification of the efficacy of these meetings must be carried out.

- c) Training and Reporting. In this final phase the objective is to close the cycle. The training department must develop the training intervention plan by taking the Causal Analysis output (arrow 10) and prioritizing with the QADatabase data. The training department will provide the development projects with the training plan and the training materials. The training department will then execute the training. Finally, at every project’s milestone, the training department will monitor the projects defects and conduct a ROI calculation. The results of that calculation will be reported to higher management.

Table 1 Process Deployment Plan summary

<table>
<thead>
<tr>
<th>Phase</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish process</td>
<td>Development and validation of the classification schema</td>
</tr>
<tr>
<td>Capability</td>
<td>Modification of the defect tracking system</td>
</tr>
<tr>
<td></td>
<td>Analysis of current defect data</td>
</tr>
<tr>
<td>Deploy Causal Analysis</td>
<td>Define a causal analysis process for the organization</td>
</tr>
<tr>
<td></td>
<td>Train the human resources in the causal analysis process</td>
</tr>
<tr>
<td></td>
<td>Include the causal analysis sessions as planned activities within the projects</td>
</tr>
<tr>
<td></td>
<td>Verification of the efficacy of the causal analysis sessions</td>
</tr>
<tr>
<td>Training and Reporting</td>
<td>Select and deploy training intervention</td>
</tr>
<tr>
<td></td>
<td>Execute the training</td>
</tr>
<tr>
<td></td>
<td>Calculate ROI and communicate results</td>
</tr>
</tbody>
</table>

8. Current State of the implementation
In order to evaluate the initial capability of the organization, we have taken a sample of the organization’s 2007 development projects. We divided the sample project’s analysts into two groups. The first group was called Taxonomy working group and the second Verification working group. The Taxonomy working group was asked to develop a classification scheme that would suit the organization’s information needs. The classification scheme is the result of the technicians’ workshops (an example is shown in Table 2). It is important to remark that the classification scheme is consistent with the organizational culture; it is domain specific and differs from the industry classifications. Then we presented the classification scheme to the second group. They were first asked to validate the classification scheme. The validated scheme was applied to their projects defects. At the end we had obtained a defect classification scheme that the organization projects can relate to. And in due course we had classified about 20% (310 defects) within the sample projects.

Table 2 Organization’s defect classification scheme

<table>
<thead>
<tr>
<th>Defect Category</th>
<th>Classification Criteria</th>
<th>ODC Category Equivalence</th>
</tr>
</thead>
</table>
| Product Integration Error | • Interface implementation do not match specification  
• Changes are not transferred to lower layers  
• Data not transferred from lower layers | Communication |
| Error in use or configuration of user interface controls | • Controls ‘Freeze’  
• Paging not working in DataGrids  
• DataGrids missing headers | Education |

The following activity was to integrate the classification scheme into the defect tracking system. A field to represent the Defect Category (Table 2) scheme was added to the defect scheme. The user interface was also modified, a combo box was added which presents the classification’s categories. In addition to this, the developer can obtain online help that displays the Classification Criteria (Table 2) for each category. The defect tracking system is a tool which is fully institutionalized within the organization, and the organization as a whole is used to receiving updates notification of the tool via email – which was also the case in this upgrade. This goes to show that the overhead to the actual development practices were kept to a minimum as was one of our goals.

In order to analyze the current data, we had to establish equivalence between the organizations classification scheme and ODC scheme (see Table 2). The equivalence rules were determined taking into account the Classification Criteria. Within our project sample, the results show that 53% of defect (classified as Education and Transcription) could have been avoided by training.

Figure 2 Defect distribution by ODC category

Phase B of the deployment plan is on course within the organization. We want to be very careful at this phase, because a causal analysis process area implementation is not required at maturity level 3, and as a result we want to be very careful in order to minimize the overhead of implementing casual analysis meetings within the organization.

As a consequence, we were asked by senior management to provide estimation of the expected benefits of our proposal. Since phases B and C have not yet been implemented, we had to estimate the process benefits in order to provide ROI results. Taking into account the defects classified in phase A, the Return on Investment of our proposal within this organization was calculated. This ROI calculation is a simulation of phase C, where organizational historical data was used to simulate the training intervention. The ROI calculation takes into account the cost of having programmers attend training, the cost incurred in the tool modification, and the cost of the causal analysis meeting. This cost was derived from the workshop with the system analyst working group.

Benefits were calculated taking into account an estimation of reduced Education or Transcription defects that should not appear after training has been applied. The organization’s historical effectiveness of training was also taken into account. As a result in the ROI calculation not all Education or Transcription defects might have been eliminated. These two previous factors...
were multiplied by the average cost of a defect. The average cost of a defect was taken from the organization’s historical data.

**Table 3 Cost and Benefits factors**

<table>
<thead>
<tr>
<th>Cost Factors</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support tool modification ( (C_{TM}) )</td>
<td>Defect reduction Number of Education Defects ( (E_D) )</td>
</tr>
<tr>
<td>Causal analysis meetings ( (C_{CAM}) )</td>
<td>Number of Transcription defects ( (T_D) )</td>
</tr>
<tr>
<td>Training ( (C_T) )</td>
<td>Average defect cost ( (C_D) )</td>
</tr>
<tr>
<td>Training Effectiveness ( (T_{EFF}) )</td>
<td></td>
</tr>
</tbody>
</table>

**Equation 2 ROI calculation formula applied**

\[
\text{ROI} = \frac{(\text{Benefits} - \text{Costs})}{\text{Costs}} \times 100
\]

With these values a ROI of 168% was calculated. The estimation also forecasts savings of 43% in terms of defect reduction costs. These results have reinforced the implementation initiative within the organization and senior management has renewed its commitment with our research work.

**9. Summary**

In this paper we presented an enhanced process for a training department. We have shown that it is possible to introduce this process with minimal impact on the organization’s current development practices. We have established a defect classification scheme within the organization that is aligned with ODC methodology. A training strategy has been designed for the training department that has proven its ability to provide positive results at the “results” level of Kirkpatrick’s model. In addition to this, by linking training and defect reduction, we have also established one of the bases of ROI calculation. Which is assuring, “beyond reasonable doubt”, that training can have impact on defect reduction.

Our current results have shown that our proposal is cost-effective; we have been able to present senior management with positive ROI values. These ROI calculations give credibility to the training department by providing visibility of the results of the training investment.

For this proposal, a positive ROI depends on the defect distribution and the associated costs. In this organization we have found that 56% of defects can be assigned to Education or Transcription. Moreover, the support tools proved to be flexible enough to allow a low cost modification.

**10. Acknowledgement**

Infocorp Software Factory supported the present work. We appreciate their continuing support with our research.

**11. References**