Organizational Learning Through the Collection of "Lessons Learned"

Joseph V. Vandeville
Northrop Grumman Corporation

vandejo1@mail.northgrum.com

"Learning is not compulsory . . . neither is survival"
-- W. Edwards Deming, Wisdom and Ignorance

Abstract

This paper provides an approach for organizational learning through the collection of "Lessons Learned." The approach focuses on organizations in the Information Technology area, but is applicable to any organization having defined processes and a mechanism for process improvement. This approach ties the lessons learned program to the process infrastructure used by the organization to collect lessons that can be acted upon by the company's process improvement program.

Keywords: Organizational Learning, Lessons Learned, Process Improvement, Capability Maturity Model, CMM

Lessons Learned as a Type of Organizational Learning

"Lessons learned" are a type of organizational learning, and although the literature abounds with examples of lessons learned, (for example, an April 2000 web search by the author for "lessons learned" using Altavista http://www.altavista.com and Google http://www.google.com each found over 130,000 pages. Yahoo http://www.yahoo.com found over 51,000 pages and Infoseek http://www.infoseek.com found over 32,000 pages.) there appears to be little information on how to collect them. This appears to be primarily an ad hoc activity that often yields only anecdotal results. Collection seems to be loosely defined, and the analysis and subsequent usage of "lessons learned" information is often lacking. Many organizations treat the recording of lessons learned as an end in itself, thereby missing opportunities to use information that is already present in the company for improvement.

The experiences of the people who actually execute a process can be one of the most important sources of input to a process improvement program. Yet valuable experience is often lost because it is not captured in a timely fashion as it is being gained, if it is captured at all. The development of a structured approach for collecting and using this information provides a mechanism to promote organizational learning by harnessing the personal and team learning that is already taking place in the organization.

Organizational Structure and Models

As a business grows, it develops and refines an organizational structure. The organizational structure is reflected in a business model, which contains the set of processes employed to produce the organization's products and services. Nevis (1995, p.73) assumes that organizations learn as they produce, so any production system can be viewed as a learning system.

In 1984, the U.S. Department of Defense established the Software Engineering Institute (SEI at http://www.sei.cmu.edu/) at Carnegie Mellon University to advance the practice of software engineering due to its criticality in defense systems. The assumption was made that organizations are made up of malleable processes that can be improved over time. A Software Capability Maturity Model1 (CMM) was developed

[1] CMM, Capability Maturity Model, and Capability Maturity Modeling are registered in the U.S. Patent and Trademark Office.

(CMM) was developed to promote the evolution of software engineering from an ad hoc, labor-intensive activity to a discipline that is well managed and supported by technology.

The CMM (Paulk, 1993) is organized as process areas containing the current best practices in the industry. The activities associated with these practices are typically non-prescriptive in that the model indicates what should be done, but leaves it up to the organization to provide the implementation that best fits its needs. The process maturity profile (CMU SEI, 2000) from the Software Engineering Institute indicates the extent to which this model has been adopted. Organizations using a well-structured process model such as the CMM can use the process infrastructure it provides for organizational learning (Vandeville, 1999).

Organizations that use process models are familiar with conducting self-examinations, typically called audits, assessments or evaluations. These activities allow the organization to measure the degree to which they conform to standards, and how well practices have been implemented and institutionalized. Often, the processes used by an organization are based on years of evolution and represent the best available at that time. Organizations that examine their current processes can not only measure compliance and institutionalization, but also use this information to determine which processes are effective and which are not.

Effective processes are those that produce the desired or expected results. Effective processes should be kept by the organization, while ineffective processes (i.e., those that do not yield the desired results) should become candidates for process improvement. The process improvement approach prioritizes the process candidates to select those that should be retired, replaced or improved.

Learning in the Process Improvement Cycle

The process improvement paradigm is efficiently characterized in the Shewart Plan-Do-Check-Act cycle (Deming, 1986, p. 88) shown in Figure 1.

The organization plans the work to be performed (Plan) by identifying the appropriate set of activities based on the organization's process model. Along with defining the sequence of activities and necessary resources, the organization will typically identify the expected outcome of the work performed, such as duration, effort involved and product quality. During the execution phase (Do), practitioners perform the work according to the predefined plans. The performance of the process and the quality of products produced are evaluated (Check) to determine whether the expected results have been achieved. If necessary, actions are taken (Act) to modify plans to either achieve the desired results or initiate an improvement.

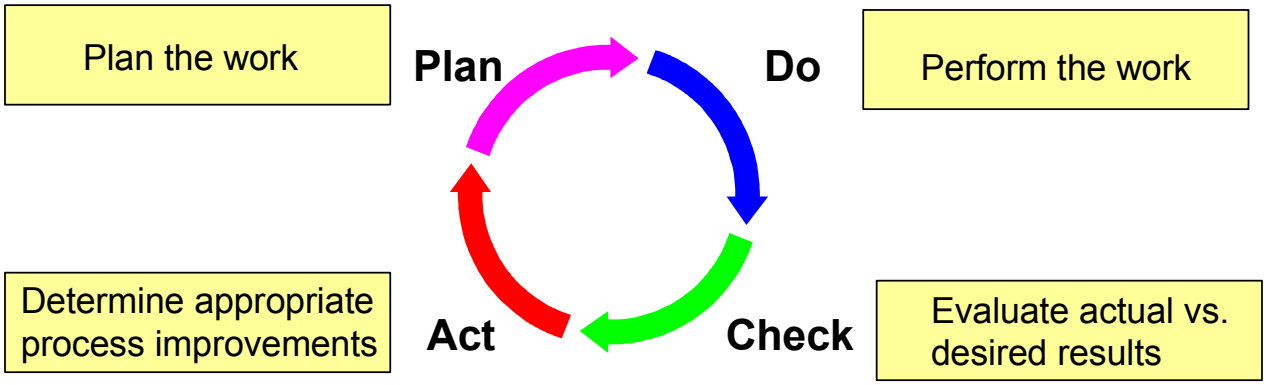

Figure 1. Shewart Plan-Do-Check-Act Cycle

Learning occurs throughout this entire cycle. When the practitioners are doing their work, they will be learning how well the defined activities are performing. This experience that represents individual or team learning can be captured as lessons learned by evaluating how well current processes are working for the practitioners. This learning can be used to promote changes to the organizational processes for use on the next project or related follow-on activity. A modification of the Shewart Cycle is shown in Figure2, illustrating the way individual and team learning becomes institutionalized as organizational learning by incorporation in organizational processes.

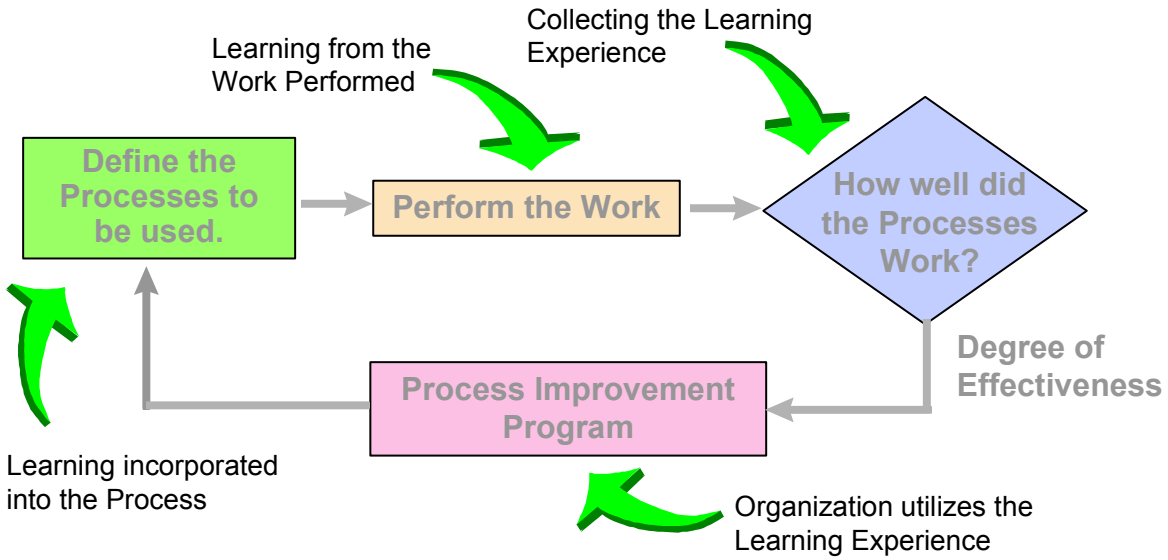

Figure 2. Modified Shewart Cycle Using the Process Improvement Program

A Lessons Learned Program

The approach presented in this paper for a lessons learned program is intended to minimize the often ad hoc approach to organizational learning by tying the lessons learned program to the process model used by the organization. This approach uses the organization's existing process framework to provide a structured, lessons learned program that can be employed during and at the completion of project execution. The assumption will be made that the organization is using the Software CMM as their model for process definition and improvement. It is further assumed that the organization is either at CMM Level 3 or at least actively working on the achievement of CMM Level 3. Level 3 is characterized as having an organizational focus.

There is an organizational way of doing business that is tailored to the needs of specific projects. The management and technical processes have been integrated to satisfy business goals, and a process group is established to promote process improvement. The framework for the proposed lessons learned program is shown in Figure 3 and described in the following sections.

Who Promotes Learning in the Organization?

One of the advantages of organizations that are at or approaching CMM Level 3 is that there is a group identified to promote process definition and improvement. This group may be known by various names and may have a mixture of full time and part time individuals. The CMM identifies this group as the Software Engineering Process Group (SEPG). Having

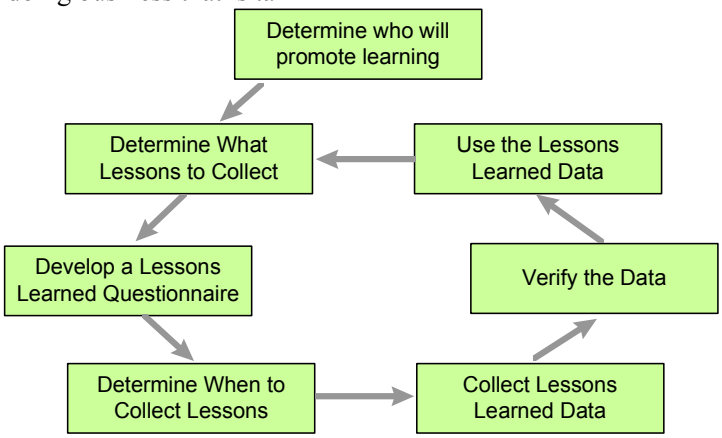

Figure 3. Framework for a Lessons Learned Program

such a group answers the question of who in the organization should learn on behalf of the organization (Miles, 1980, p. 49).

This group typically has an "organizational focus" and can better apply learning to achieve organizational results than practitioners who typically have a "project focus." This organizational focus will allow a better association of the results of learning with organizational behavior. Furthermore, the process group will have the time needed for organizational learning. Often project personnel will be too tied up in meeting project schedule demands to have the time to analyze and use the lessons they learn to benefit the organization.

Determining the "Right" Lessons to be Learned

It is critical to collect meaningful lessons learned information so that subsequent analysis will be productive. Often organizations leave it up to the staff to come forward with lessons learned. This lack of a collection infrastructure can lead to poor results. It may result in no information being captured, or the information captured may be relatively unusable.

Examples of unusable data are those that are outside the scope of the process improvement program. For example, a lesson learned may be that the staff was underpaid. Often the process improvement program does not address human resource issues, and therefore cannot effectively address this type of perceived problem. Other types of issues may involve computer resources (number of computer terminals, speed of printers, maintenance of equipment), the workspace environment (heating, cooling, lighting, ventilation), and inter-personnel conflicts.

Once such perceived problems are reported, there is a certain expectation that something will be done. When the process improvement program cannot take any meaningful action, there may be a loss of confidence in the effectiveness or validity of the lessons learned system. Furthermore, organizations often have departments that can address the types of issues that are outside the process improvement program. These may include a Human Resources Department and an Internal Information Systems Department.

Use of a Questionnaire to Collect Lessons Learned Data

Instead of waiting for staff to come forward with lessons learned or simply asking what they have learned, a proactive approach using questionnaires is proposed. Using questionnaires offers several benefits, such as:

- The collection of lessons can be focused on specific items.
- The level of detail to be collected can be pre-determined.
- The collection can be employed at appropriate times in the project life cycle.

Using a questionnaire not only provides a focus on what is to be collected, but also how it is collected. For example, collecting information as yes-no answers to questions has an advantage in being easy to evaluate. However, these types of questions and answers may be artificial in the sense that they do not capture the diversity of opinions that may exist, and instead compress them into discrete yes-no responses

The author’s experience in conducting assessments has shown that engineers and technical staff are often uncomfortable with yes-no or true-false questions, since they recognize that their practices often include shades of gray. Consider the question “Do you manage requirements?” Some people will have little problem with such a question. If they are optimists and perform requirements management to any reasonable degree (by their definition), they will say “yes." Others who are pessimists may say “no” if there is even the slightest aberration with the requirements management process. Either of these two extremes will distort the data.

Other people will have difficulty with a simple “yes” or “no” answer, recognizing that requirements management is a practice complex and rich in attributes. They will attempt to establish a discriminator function, based on their beliefs to help determine the answer to the question. Again the data will be distorted since different people will have different discriminators, which they may be unwilling or unable to articulate. The interviewer may attempt to define a discriminator function, but it may be very difficult to determine what differentiates requirements management from non-requirements management.

If the results are to be used for process improvement, it is important to determine the degree to which requirements management is practiced. This will allow a process engineer to determine if resources should be expended on improving this process.

Since the information collected is the result of a human activity, it is likely to be incomplete, imprecise, uncertain and ambiguous. This makes it difficult to impose much formality on the collection. People will feel intimidated by being asked to quantify essentially qualitative information. Most people will feel uncomfortable in answering a question such as “On a scale of 1 to 5, rate how well you managed your requirements." A more natural question can be posed using a Likert Scale. This type of question might be “How well do you feel

requirements were managed: very poorly, reasonably well, about average, better than average, very well." Such questions will appear less threatening and are likely to yield better results.

Lessons Learned vs. Process Assessments

For the lessons learned to be useable by the process improvement program, the collection mechanism should focus on the key processes used by the organization. The same processes are also examined during process assessments. There are however, fundamental differences between process assessments and collecting lessons learned, although both depend on having a well-defined process model.

During process assessments, the auditors are attempting to determine that the implementation of a process satisfies the requirements of the process model or standard. The assessment normally occurs in two parts: a sufficiency audit and a compliance audit. The sufficiency audit is to determine that processes are in place, which are sufficient for the needs of the business. The compliance audit determines that these processes are actually followed.

The lessons learned program in contrast, assumes that processes are in place and there is some measure of compliance. The goal of the lessons learned program is to determine how well the practices are producing the desired results for the organization. For this reason a certain degree of process maturity is needed for a lessons learned program to be effective. If there is no consistency in the execution of a process, then learning about its use will have little value since the process is unlikely to be repeated. A lessons learned program therefore, should be viewed as a complement to the assessment program.

Developing a Lessons Learned Questionnaire

Questionnaires used by the assessment process can be used as a starting point for developing lessons learned questionnaires. As described above, the assessment questionnaire is seeking to determine if a practice is in place and used. The lessons learned questionnaire, on the other hand, will assume that the practice is being used and will seek to determine how well it is working.

The questionnaire provided by the Software Engineering Institute (SEI) for use in software CMM assessments is a good example of a process assessment questionnaire (Zubrow, 1994). It is used to give auditors a rough approximation of an organization's level of process maturity.

The following steps can be used to develop an initial lessons learned questionnaire from an assessment questionnaire:

1. Determine how the activities addressed by the assessment questions are performed. Note that this will be organization and perhaps even project specific.
2. Determine the activities the organization wishes to learn about.
3. Compose new questions about the activities performed to learn the practitioner's perceptions of the results achieved by executing the practice.

As an example, the CMM questionnaire used for assessments asks the following question in the area of requirements management: "As the systems requirements allocated to software change, are the necessary adjustments to software plans, work products, and activities made?"

The organization may perform the following activities when software requirements change:

1. Repository of software requirements is updated to reflect the changes.
2. Software development plans are modified.
3. Related design, code and test products are changed.
4. Resource allocations (staffing, labs, etc.) are adjusted.
5. Changes are communicated to the necessary staff.

In this example, the organization may wish to know how well the software requirements repository was maintained, how well resources requirements were adjusted, and how well the changed requirements were communicated to the necessary staff.

The lessons learned questionnaire may consist of the following:

1. How well were changes to requirements maintained? [very well, quite well, reasonably well, not well, hardly at all]
2. How well were staffing resources adjusted to compensate for the changes in requirements? [very well, quite well, reasonably well, not well, hardly at all]
3. How well were changes in the software requirements communicated to the necessary design staff [very well, quite well, reasonably well, not well, hardly at all]

Determining When to Collect Lessons

Consideration needs to be given to the appropriate collection time for lessons learned. Collection can be done at the com-

pletion of a project to record and preserve the learning achieved by the staff during project execution. These data would then be used by subsequent projects, typically in their planning phases.

A better approach is to collect lessons at the completion of a development phase or other significant event, such as a project milestone. For example, if a requirements review is conducted at the completion of the requirements analysis phase, this would be a good opportunity to capture lessons that the team learned in performing this task. This will ensure that the staff experiences are still relatively fresh. The collection of lessons learned can be part of the exit criteria for each life cycle phase. Similarly, lessons learned on previous projects can be part of the input or entrance criteria of life cycle phases on current projects. In this way organizational learning will be propagated from project to project.

Some types of lessons can be collected periodically. This is especially true of processes that are independent of development phases. For example, once initial requirements have been baselined, changes are managed throughout the entire life cycle. Depending on the frequency of requirements changes, lessons can be collected at periodic time intervals to determine how effective the requirements management process is working.

Collecting and Verifying Lessons Learned Data

The collection of lessons learned will probably never be an exact science. Therefore, some degree of verification of the collected data is needed to give credibility to its use in process improvement. Statistical methods such as population sampling can be used to give confidence that the information collected represents the population as a whole.

Another form of verification is to obtain information from relatively independent sources. For example, asking the same question of management and of the workforce will often give different views of the same work performed. These results can be used to corroborate one another. Also in a team environment such as an Integrated Product Team (DoD, 1996), asking the same question of different disciplines on the same team will allow verification of responses.

The Integrated Product Team structure offers many interesting opportunities to collect meaningful data since often customers and suppliers of products work together on the same team. For example, the systems engineer may say that the system requirements allocated to the software portion of the project were well known at the beginning of software design, whereas a software engineer may say that they were not. Although such responses may not clearly indicate whether or not the software requirements were well known, they do indicate that further investigation is needed into the effectiveness of the communication mechanisms between related organizational disciplines.

On a cautionary note, care must be taken to ensure that the collection process is sensitive to individual privacy. Collecting data about practices, not individuals, will help ensure that unbiased answers are given during data collection.

Analysis and Use of Lessons Learned Data

The process group responsible for the definition and improvement in the organization can use the collected lessons learned data to evaluate how effective the organization's processes are in producing the desired results. Since the lessons learned are tied to specific processes and are ordered by the degree to which practitioners perceive them to be functioning (e.g., very well through not very well), the process group can use this ordering to prioritize the initiation of process improvements.

Those lessons that indicate processes are functioning well can be left alone or at least need to receive only minimal attention. Processes that are not working well should be examined to determine the cause for the perception of inefficiency. Those processes that are determined to be inefficient and are considered critical to the success of the organization should receive high priority for process improvement.

The process group will have the organizational resources and process knowledge to determine the causes of process inefficiency. With a process improvement plan in place including prioritized proposed process improvement initiatives, management can determine how best to allocate resources (e.g., staff and budget) to achieve their goals.

Summary

By basing the collection of lessons learned on the process models used by the organization, a lessons learned program can be established which extracts information that can be more precisely mapped onto the organization's process improvement model. This approach will tend to eliminate anecdotal information that is often of little use in process improvement. It also allows the organization's existing process improvement infrastructure to be employed to advance and utilize the results of organizational learning. It also supplements existing process assessments, to determine how well implemented practices are satisfying organizational needs.

References

Carnegie Mellon University, Software Engineering Institute (2000). "Process Maturity Profile of the Software Community 1999 Year End Update." http://www.sei.cmu.edu/sema/profile.html.

Deming, W. E.(1986). Out of the Crisis, Massachusetts Institute of Technology, Cambridge MA

Department of Defense, DoD Guide to Integrated Product and Process Development, Version 1.0, 1996, Office of the Under Secretary of Defense (Acquisition and Technology), Washington, DC.

Miles, R. H., Randolph, W. A. (1980). "Influence of organizational learning styles on early development", The Organizational Life Cycle , Kimberley, J. H. & R. H. Miles (eds.), San Francisco, Jossey-Bass, pp. 44 - 82.

Nevis, E. C., DiBella, A. J., Gould, J. M. (1995). "Understanding Organizations as Learning Systems," Sloan Management Review, Vol. 36, No. 2, Winter 1995, Pp.73-85.

Paulk, M. C., Weber, C., Garcia, S., Chrissis, M., Bush M. (1993). *Key Practices of the Capability Maturity Model, Version 1.1*, CMU/SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. http://www.sei.cmu.edu/publications/documents/93.reports/93.tr.025.html

Vandeville, J. V., Shaikh, M. A. (1999). "A Structured Approximate Reasoning Based Approach for Gathering 'Lessons Learned' Information from System Development Projects", Systems Engineering, The Journal of the International Council on Systems Engineering, 1999, Vol. 2, No. 4.

Zubrow, D., W. Hayes, J. Siegel, D. Goldenson, (1994) "Maturity Questionnaire," June 1994, CMU/SEI-94-SR-7, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA. http://www.sei.cmu.edu/publications/documents/94.reports/94.sr.007.html