



The Need for a Useful Lessons Learned Database

George Jackelen
EWA Inc.

Literature and standards mention the importance of lessons learned. So why are we still having problems? Why have we not learned our lesson?

Scene: “The project is finally over,” the project manager muses. “Everyone has either been transferred, is hanging around to wait for the next assignment, or is satisfying the old saying, ‘The job ain’t done until the paperwork is finished.’”

“Hmm. That just leaves preparing the final costing reports, archiving the products and related paperwork in case there is a post-audit, and preparing lessons learned. Lessons learned should only require a page or two; I’ll find out who is available.”

An Afterthought

As shown by the above example, documenting lessons learned is often downplayed as a needed, and sometimes required, task to help future projects avoid problems. The need for lessons learned is described as:

“... so that they [causes of variances, the reasoning behind the corrective action chosen, and other types of lessons learned] become part of the historical database for both this project and other projects of performing organizations” [1].

So, what is the problem? My experience in the federal government and industry is that the current lessons-learned efforts are not seen as worth the time spent in their implementation. Lessons learned are mainly afterthoughts done at the end of a project to close out one of the final checklist items to terminate a project. Providing lessons learned when they occur is considered to be time consuming, and there are things of higher priority that must be completed first. There is a prevailing view that the idea of having lessons learned is good. In reality, presented issues are normally typical (for example, schedule and cost must be managed) or unique to a project.

As has been stated by many authors and project managers, “It is generally cheaper to prevent problems than to fix problems.” This is especially true of project problems. If done correctly, an organizational lessons learned database can prevent project headaches and save time and money.

Current Lessons Learned Task

What causes these obstacles? Let us take one example, “What is wrong with looking at old problem reports?” Problem reports do not provide lessons learned information since management and business issues are normally not part of problem reports. Also, many problem report issues are so technical and project specific that they do not help future projects anticipate or avoid problems. As a result, many problem-report solutions are also too technical and project specific. In addition, most problem reports do not address the question, “What could have been done to prevent this problem?”

Proposed Lessons Learned Task

To overcome these obstacles and to make a lessons learned task work, there must be a clear understanding of the purpose for documenting lessons learned. I propose the following definition:

Lessons learned task—An ongoing project task to document a project’s major negative and positive issues. The documented lessons learned are used to prevent issues from having a negative impact on a project and to provide alternative ways of doing things.

To implement a continuous lessons learned task, without making it an overhead nightmare, it is necessary to define the type of issues to be continuously reported. For example, issues approaching a plus or minus 5 percent impact on cost, schedule, or deliverable product size or performance shall be entered into an organizational lessons learned database.

A second lessons learned subtask is to consider the format of a lessons learned report. A problem report identifies a problem, the cause, and solution; a lessons learned report identifies the same things as well as what should have been done in advance to recognize and prevent the issue.

A third lessons learned subtask is that a successful lessons learned database needs to have the assignment of responsibilities. Someone in a project management office could be responsible for cost, schedule, customer interactions, and management lessons learned. A quality assurance organization could be responsible for deliverable product issues (including processes, development, delivery, installation and maintenance).

Thus, a lessons learned task does not have to be complicated, costly, or wait until project termination.

Is It Worth the Effort?

A timely lessons learned database must be part of an organization’s process improvement effort. This database can be used to reduce risks by providing users with information about how people were able to recognize risks (hopefully in advance) and overcome these risks. As a result, a lessons learned database becomes not just a project tool, but also an organizational tool to help ensure the past can be used to help the future of an organization. An organizational lessons learned database can economically reduce risks and costs while providing greater benefits than independent project lessons learned databases.

Recommendation

A common statement is that quality must be built into a product or service. For product quality to occur, a person/ organization must step back and realize that quality must be built into a

project before quality can be built into a product or service. Going a step further, quality must be built into an organization before quality can be built into a product. The following steps can help establish quality projects:

- Define the format and criteria for information to be stored in an organizational lessons learned database.
- Assign responsibility for timely updates of the lessons learned database to a few people or organizations. The responsibility should include examining an organizational lessons learned database prior to and during a project.
- Identify metrics, including frequency of measurement and reporting, for ensuring effective lessons learned task implementation and to measure the level of success. For example, have project metrics to identify the lessons learned database's usefulness. This could include such questions as:
 - What issues were prevented or resolved, based on the provided information?
 - What information is being duplicated? Does the organization have a bigger problem?
 - Is the lessons learned database periodically audited to ensure it is being updated with timely information that is used?

Conclusion

A lessons learned database is a useful management tool to help projects reduce the number and level of risk items and to provide useful information to identify positive ways to run a project. The main need is for a positive attitude about the value and cost effectiveness of efficiently run lessons learned databases. In addition, there can be a tremendous organizational benefit to reduce risk and to assist in a process improvement effort. There is a need for useful lessons learned databases. ♦

Acknowledgement

I want to thank Dan Solomon for his advice.

Reference

Project Management Institute. *A Guide to the Project Management Body of Knowledge*, Project Management Institute, Upper Darby, Pa., 1996.

About the Author



George Jackelen is project manager and analyst for two NASA Independent Verification and Validation projects. During his 30-plus years experience, he has performed software and hardware quality assurance for DoD and industry, and developed or provided review comments on ISO, IEEE, DoD and contractor standards, policies, procedures, etc. He has a master's of science degree in Computer Science from Texas A&M University.

1000 Technology Drive
Fairmont, W.Va. 26554
Voice: 304-367-8252
Fax: 304-367-0775
E-mail: gjackele@ewa.com

Coming Events 2000

February 28–March 3

16th International Conference on Data Engineering ICDE 2000
<http://www.research.microsoft.com/icde2000/>



March 6-8

13th Conference on Software Engineering Education and Training CSEET&T
<http://www.se.cs.ttu.edu/CSEET2000>

March 6-10

Software Management/Applications of Software Measurement SM/ASM 2000
<http://www.sqe.com/smasm/2000/>



March 13-17

6th International Conference on Practical Software Techniques
<http://www.softdim.com>

March 20-22

5th Annual Association for Configuration & Data Management Technical Conference
<http://www.acdm.org>

March 20-23

12th Software Engineering Process Group Conference



<http://www.sei.cmu.edu/products/events/sep/2000/>

April 11-14



<http://www.infosec.co.uk/page.cfm>

April 18-20

FOSE :Leading-Edge Technology for Leaders in Government
<http://www.fedimaging.com/conferences>

April 30–May 4

The 12th Annual Software Technology Conference



<http://www.stsc.hill.af.mil/index.asp>

June 20-22

Product Focused Software Process Improvement PROFES 2000
<http://www.ele.vtt.fi/profes2000>

August 28-31

First Software Product Line Conference SPLC1
<http://www.sei.cmu.edu/plp/conf/SPLC.html>