

# The Good, the Bad, and the Ugly

## Lessons Learned From Them All

Eugene A. Razzetti



**N**OTHING IS MORE COST-EFFECTIVE THAN GETTING EVERYTHING YOU CAN OUT OF THE FUNDS you already have spent.

Let that sink in. Your program's funding already is identified, obligated, and being spent. You will get a product at the end. But will it be worth what you spent—or could you have built it cheaper, better, or faster?

Don't wait too late to find those answers, or to identify a nonconforming product or service that could have been corrected during development or manufacture. When the project is done, you may have no choice but to back-fit or modify—or throw it away entirely. Collecting lessons learned is too important to be left until the end. By then it is too late for that program and probably for your next one as well.

Some years ago, I was one of a team of subject-matter experts commissioned to analyze maritime operations during Operation Iraqi Freedom. The operation had been completed by then, and our sponsor told us to “report what we did right.” No lessons-learned data had been corrected until then. We were forced to reconstruct tracks, sift through thousands of e-mails—barely 10 percent of which were relevant—and interview mariners long after their reassignment. Worst of all, no input had ever been solicited from the Warfighters, many of whom jumped at our invitation to participate in the analysis.

In the end, we analyzed what we wanted, interviewed whom we wanted, and created our own set of questions and metrics. We reported the “Good,” and there was plenty of it. However, we also reported what was “Bad” and what was “Ugly.” It is good to be the subject-matter expert.

---

**Razzetti**, a retired U.S. Navy captain, is a management consultant, auditor, military analyst, and frequent contributor to *Defense Acquisition magazine* and the former *Defense AT&L magazine*. He is the author of five management books, including *Fixes that Last—The Executive's Guide to Fix It or Lose It Management*.

Lessons learned are findings and experiences distilled from a project, lessons that should be actively taken into account not only in future projects, but for “mid-course corrections” to current projects. Findings may be positive, as in a successful test or mission—or negative, as in a mishap or failure.

The two most serious mistakes that a program manager can make to render lessons learned ineffective (read: meaningless) are (1) not to include Warfighter/user inputs; and (2) to wait until the end of the program to collect them.

Regrettably, capturing lessons learned is usually thought of as a standalone, static, act—awaiting conclusion of the program or exercise. Furthermore, the end-product, once collected, often enjoys a late launching, a limited distribution, and (worst of all) a lack of distribution to participants whose contribution would likely be uncomplimentary or disapproving.

A dynamic and ongoing strategy of collecting lessons learned in real time is vastly more meaningful than collecting them at the end of the effort. Collecting lessons learned after the fact, and with all the damage done, is more correctly called an investigation. Nobody likes investigations.

It can be difficult to keep track of lessons learned in a timely manner without a formalized strategy. Learning a lesson about a program has, assuredly, the potential to improve the next program. Equally important, however, is the potential for an existing program to improve itself.

A robust and ongoing program of capturing lessons learned is essential to dealing with (but not limited to):

- Risk identification and prioritization
- Best practices
- Design issues
- Test planning and revision
- Budget and quality plans
- Delivery schedules
- Manpower and training requirements

Risk identification and prioritization form the basis for developing, validating, or revising best (management) practices. Design issues can come from the factory floor, the battlespace, or any place in-between.

Figure 1 describes an ongoing approach to capturing lessons learned in a timely manner.

Again, effective, actionable, lessons learned must be captured and analyzed in real time, with corrective actions initiated as soon as possible. They cannot be exclusively the subject of after-action reports or “hot washes.” Find the problem, apply an appropriate corrective action—and then follow up to assess the effectiveness of the corrective action. Meaningful after-action reports (by whatever title) will state the findings, but should also state the corrections, and the (measurable) effectiveness of those corrections.

### Start With the Project Plan—Everything for Lessons Learned

You need to have at your disposal: the program goals and objectives; metrics; feedback loops/status reports; and user/Warfighter input. Get all the input as soon as possible—go over everything, and remember that gathering lessons learned is a continuous process.

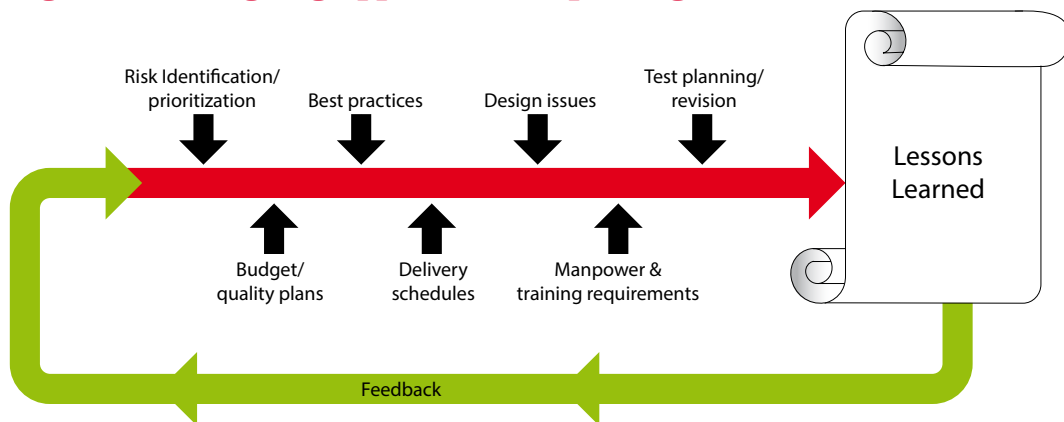
Start collecting data at the beginning and keep collecting throughout the program—don’t wait until the end and try to reconstruct. Problems areas need to be addressed immediately. Eagles recognize that—ostriches don’t.

Table 1 is taken from a previous article of mine applying program management requirements first to System Integration, and then to Expeditionary Warfare and Battlespace Management. Those same requirements appear again in Table 3. The requirements are generic, and the reader is welcome to substitute as appropriate.

### Tracking Lessons Learned

Table 2 includes an effective means of weekly tracking lessons learned, using an easy-to-update Excel spreadsheet. The preparer inserts the date at the top and the findings, with tracking numbers assigned. An arbitrary risk value is

**Figure 1. An Ongoing Approach to Capturing Lessons Learned**



Source: All figures and tables by the author.

assigned, along with warnings, causes, recommended corrective actions; responsibilities, and milestone dates.

Revision takes minimal time, and can come from numerous sources.

The completed document may remain in electronic form throughout its life—on a computer screen and possibly on a projector. Trend graphs (e.g., of welders qualified) can be updated and attached for better understanding and tracking. There may be no need for hard-copy distribution.

**Table 1. Getting the Material for the Lessons Learned**

Requirement	Program Mgmt.	System Integration	Expeditionary Warfare	Battlespace Mgmt.
Mission planning; concept development	✓	✓	✓	✓
Design/development (including hardware/software)	✓	✓	✓	✓
Modeling/simulation	✓	✓	✓	✓
Research and development	✓	✓	✓	✓
Risk management plan	✓	✓	✓	✓
Gap analysis	✓	✓	✓	✓
Core team developed; responsibility/accountability assigned (including decision makers)	✓	✓	✓	✓
Warfighter involvement	✓	✓	✓	✓
Performance-oriented; metrics developed/consistent/actionable	✓	✓	✓	✓
Test plans developed; tech yield identified	✓	✓	✓	✓
Contract in place; executable	✓	✓	✓	✓
Connectivity/feedback	✓	✓	✓	✓
Configuration/change management process defined/in place	✓	✓	✓	✓
Internal/external security procedures in place	✓	✓	✓	✓
Life-cycle management plan	✓	✓	✓	✓

**Table 2. Tracking Lessons Learned on a Weekly Basis**

(Example, using an EXCEL spreadsheet)

Nr.	Event	Risk	Early Warning Signs	Cause	Required Actions(s)	Materiel/ Non-Materiel	Responsibility	Completion Date	Status
C-1	HP welds		Failed welds; slag	Inadequate welder training/qualification	Materiel	Prod. Dept.	Jan. 31	Qual. Prog. Revised	
C-2	HP fittings		Late arrival on-site	Vendor delays	Revise contract; replace vendor	Non-Materiel	Prod. Dept.	Dec. 31	Bidding process begun
C-3	Safety hazards		Workplace accidents	New/untrained employees	Conduct safety training; Recertify safety equipment	Non-Materiel	Safety/HR	Ongoing	In progress

Prepared:	J.Smith	Reviewed:	<i>J.P. Jones</i>
Date:	Oct. 25	Date:	Oct. 26

The document not only forms the source for the monthly report, but for any other actions requiring a credible (positive or negative) justification.

Table 3 combines Figure 1 with Table 2 (again, readers can substitute their own inputs). We now have created a dynamic lesson learned “management tool” where perhaps only an “after-action report” had existed.

A lessons-learned database spreadsheet model is a multi-project directory that you can edit to fit your program’s needs. The model acts as the storehouse for all of your lessons learned across projects and throughout the program’s life cycle. It can be used as reference as you start new projects, as your team looks for process improvements to make throughout the program, and to document successes and recurring issues in the projects as they happen.

To use this monthly report, just input the material from the weekly report, and add supporting information about the project, and any other relevant notes. Once you have input the lessons, you can filter and sort using the process categories and project type. You could even remove or add categories for simpler or more comprehensive sorting capabilities. Encourage your project team to exploit this model, add their own insights, and reference the lessons on future projects.

**The Good—the Successes: What Went Right**

“Lessons learned” often refers to failures and needed improvements to a team, a piece of equipment, or an operation. But it is just as important to capture the successes—but not just to celebrate. Successes often identify or reinforce best practices for the remainder of the program,

**Table 3. Monthly Plotting of Lessons Learned With Program Management Requirements**

(Example, using an Excel Spreadsheet)

Program: Pressure Vessels	Monthly Lessons Learned Review (not completed)							
Program Requirements vs. Lessons Learned	Risk identification/prioritization	Best practices	Design issues	Test planning/revision	Budget/quality plans	Delivery schedules	Manpower & training rqmts	Comments
Mission planning; concept development								
Design/development (including hardware/software)								
Modeling/Simulation								
Research and development								
Risk management plan								
Gap analysis								
Core team developed; responsibility/accountability assigned (including decisionmakers)								
Warfighter involvement								
Performance-oriented; metrics developed/consistent/actionable								
Test plans developed; tech yield identified								
Contract inplace; executable								
Connectivity/feedback								Not established
Configuration/change management process defined/in place								
Internal/external security procedures in place								
Life cycle management plan								
Prepared:	Date:							
Reviewed:	Date:							

future programs, projects, and processes. The (usually) expensive and hard-won knowledge should be shared. Everyone should learn and benefit.

Whether the success came from a short-term process change, a commitment to existing process, communication tweaks, or something else, it is worth documenting this to realize what “works” and can and should be carried on to the next project for you work and your team.

**The Bad—the Failures: What Went Wrong**

By “Bad” failures, I am thinking of materiel failures, which can come in the form of (to name a few):

- The inability of a product/system to achieve its desired results
- A higher-than-expected failure or breakdown rate or a diminished life cycle
- Problems with system/sub-system integration
- Unsafe product or system operation
- An unacceptable environmental footprint and/or a reduced product life cycle

Program managers and their bosses need to remember to fix the problem, and not to just fix the blame. The process is not punishment—it is management. A failure of any dimension must bring with it a realistic, actionable, solution—plus the assignment of a responsible individual or organization and a mechanism for assessing the effectiveness of the corrective action. Metrics and milestones are vital. Lives may depend on them.

**The Ugly: What We Missed**

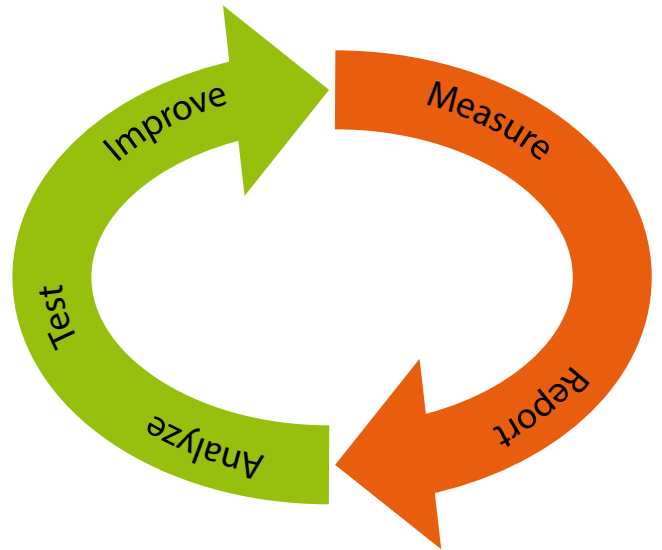
“Ugly” means non-materiel failures that probably never should have occurred. It has been said: “Beauty is only skin-deep, but ugly goes right to the bone.” So, what is an “ugly” failure? Here are a few examples:

- The program is over budget
- The contract was awarded to the lowest bidder, period
- The contract was, in some form, unexecutable
- There is no funding fenced for training/simulation
- Milestones are unrealistic or meaningless
- There are no meaningful metrics.

**Continual Improvement**

Continual improvement is (or should be) the mainstay of any modern quality management system, especially in defense acquisition programs, as we commit to improving mission effectiveness by improving our product quality, employee communication, work environment, resources, and user satisfac-

**Figure 2. Continual Improvement**



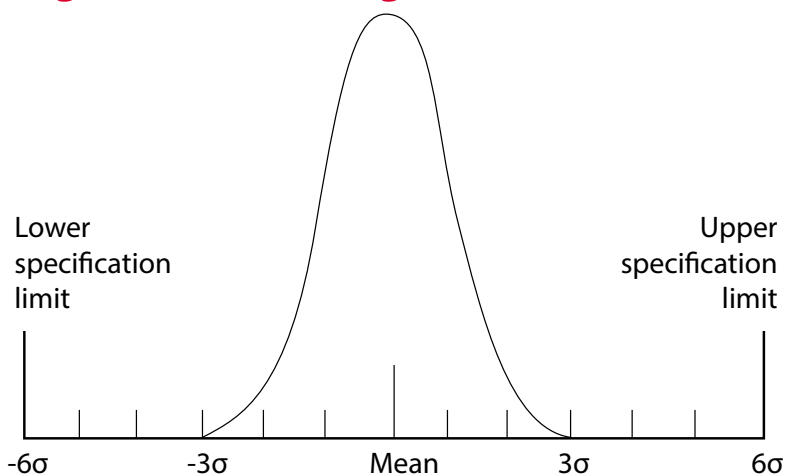
tion. All levels of the organizations participate; warfighter involvement is sought after and optimized.

Senior management provides the leadership, support, and resources to make continual improvement a priority. Middle management and process owners focus efforts and resources and review progress, and employees identify opportunities and recommend improvements.

Figure 2 depicts the constancy of continual improvement. A continual improvement “mindset” drives the lessons-learned strategy for successful program management

Ongoing data analysis, such as Six Sigma described in Figure 3, can provide significant and actionable information on operational performance and improvement opportunities. Program managers review data, make decisions, and

**Figure 3. Generic Six Sigma Presentation**





act on the findings provided by the data. Performance data, collected and analyzed with an Excel spreadsheet model, can provide real-time lessons learned, especially when the contract calls for manufacturing large quantities of precise products (e.g.; small arms ammunition).

## Summary

### **The only mistake in life is the lesson not learned.**

**—Albert Einstein**

Capturing lessons learned is too important to be left until the end. By then it's too late for that program and probably for your next one as well. Lessons learned should be captured, benefits and impacts analyzed with reference not only to the current program, but to future programs as well. You need to start at the top, with policies, procedures, and system steps—where both the positive and the negative are applied across a broad spectrum of government acquisition programs.

No process or product is so complex that it cannot be modeled and analyzed. In fact, the greater the complexity the greater the need for analysis. If you need a database, make one, analyze it, and take action on what you find. Use all the tools at your disposal, including Excel and Six Sigma. The best management practices that you create (or validate) in the lessons learned process confirm that you have identified the risks, and have done what you can to mitigate them.

The two most serious program manager mistakes that can make ineffective the lessons learned are to not include the participatory input of Warfighters and users and to wait until the end of the program to collect them.

Every program will have its “good,” “bad,” and “ugly,” Identifying and minimizing the ugly will minimize the bad, and (in doing so) will maximize the good—and the earlier the better.

*The author can be contacted at [generazz@aol.com](mailto:generazz@aol.com).*



## **DTIC Online** *Supporting the Warfighter*

### **One Site One Source**

for Scientific and Technical (S&T) Information  
*The single entry-point to the Department of Defense S&T information resources*

Use DTIC to:

- Access almost 1 million technical reports
- Query MultiSearch to retrieve information from more than 50 S&T resources in one search
- Download the Congressional Budget 24 hours after release
- Request an experienced Reference Librarian to assist with your research or exercises

**Visit  
[www.dtic.mil](http://www.dtic.mil)  
today!**