25 Valuable Lessons I Learned as a Systems Integrator and Some I Didn’t

Scott W. Sommer, P.E., C.A.P.
Automation Technology Manager
Jacobs Engineering Group, Inc.
Conshohocken, PA 19428

Christopher Russell
Principal Process Control Technology Specialist
Bayer Technology Services Americas
Pittsburgh, PA 15205

KEYWORDS

Systems Integrator, Systems Integration, Automation, Lessons Learned, Project Execution, Quality Improvement

ABSTRACT

Systems Integration is tough work. Often someone else’s design must be implemented by a systems integrator to the complete satisfaction of the owner. If this isn’t difficult enough on its own, the systems integrator is usually not only under a schedule crunch they did not create, but also becomes “critical path” at some point in the project. Still, systems integration is rewarding work, and tens of thousands of professionals are engaged in this field.

All systems integrators strive for perfection. Some actually achieve it on a project-specific basis, but every new project brings new challenges to overcome. The success of the project may depend more on how the systems integrator overcomes those challenges than on the technical abilities of the integrator’s project team. Every project produces a list of “lessons learned”; the challenge after the project is complete is to learn from those lessons as to not repeat the same mistakes.

Each of the authors has been engaged in systems integration for over 20 years. The authors have experiences from being on “both sides of the fence”, as one has previously owned his own systems integration firm. Based on these experiences, and from interviews and surveys conducted with dozens of systems integrators and end users, a useful list of these lessons has been compiled. By sharing these lessons with the industry, the hope of the authors is that systems integrators, end users, and managers alike would not only gain respect for the role of systems integration in industrial project execution, but also avoid pitfalls and errors that keep a good project from becoming an excellent one.

Presented at ISA EXPO 2007, 2-4 October 2007, Reliant Center, Houston, Texas

This paper and presentation will cover 25 of the most common lessons learned from past systems integration projects. The power of this list is not in its compilation, but in the knowing where the pitfalls to project most commonly lie. From the authors' own experiences, some lessons are harder to learn than others, and some are repeated unknowingly even after 20 years. These lessons will be presented along with steps to avoid falling into them. Every systems integrator will benefit from this knowledge, and just perhaps, each will be able to avoid all of these "land mines in the path of success".

INTRODUCTION

Lessons Learned: an industry buzzword. Everyone reading this paper has probably participated in a Lessons Learned session during their career. We do it as automation professionals. The U.S. Army does it. Large corporations do it. Mom and Pop manufacturers do it. Why not? It is the best way to share the common mistakes and successes encountered during a project with a larger audience. The goal: to learn the root causes behind, understand the implications of, and application of these lessons with the intent to improve quality, performance, and customer satisfaction.

Books, seminars, courses, software products, and a host of other media have been developed to address the recording, analysis, application, and archival of Lessons Learned. This paper will make no attempt to put forth a methodology for doing these tasks. This paper is dedicated strictly to the dissemination of the most prevalent Lessons Learned from millions of effort-hours[1] spent by Systems Integrators world-wide.

This paper will present these lessons from two vantage points. First, the most common Lessons Learned from the authors' experience in systems integration will be presented, supplemented by results of a widely-distributed survey [1] hosted by the authors. Secondly, the most common Lessons Learned from the authors' experience as end users to Systems Integration work will be shared, also supplemented by results from the same survey.

The survey was conducted in June, July, and August, 2007, and was sent to over 500 systems integration professionals. A combination of distribution lists, personal contacts, and recognized industry experts were invited to participate in the survey. Over 200 individual responses were returned, with the following demographics recorded from the respondents.

Currently Employed as a Systems Integrator	69.3%
Formerly Employed as a Systems Integrator	8.0%
End User Actively Managing Systems Integrators	22.7%
Not a Systems Integrator or End User	0.0%
	100.0%

Table 1. Classification of Respondents

[1] With 200 responses from the authors' survey and the demographics of these respondents, this number has been assumed, not validated. However, just 100 engineers working 5 years each equals one million effort hours. It is safe to say collectively, the respondents of the survey, used as the basis of this paper, have spent at many millions of effort hours in their SI careers.

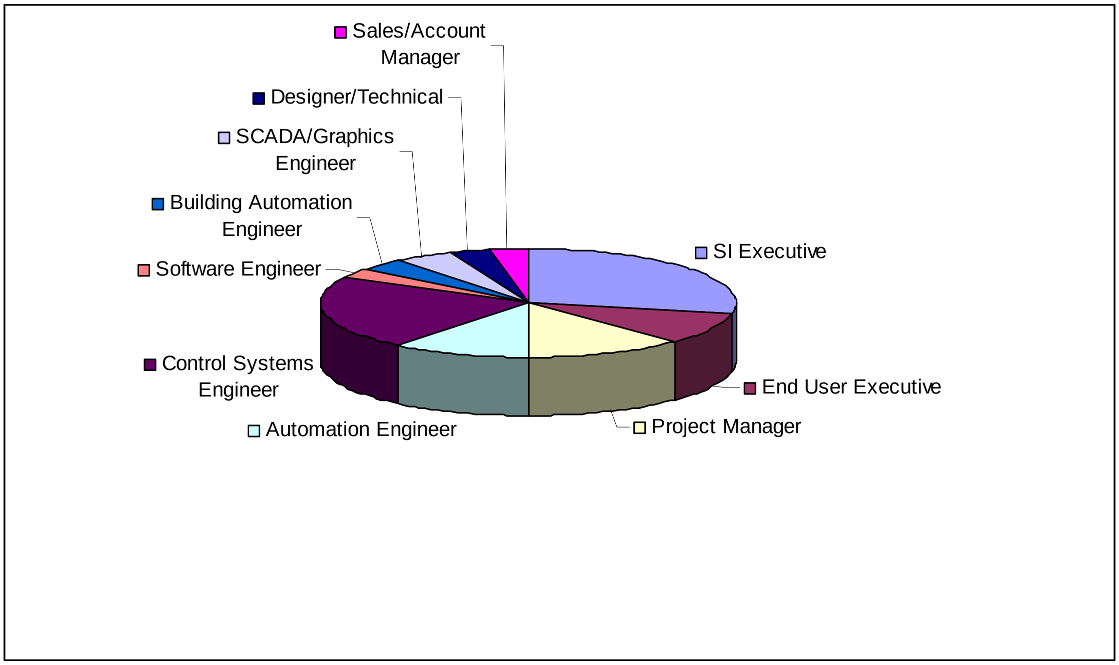

Figure 1. Primary Job Function of Respondents

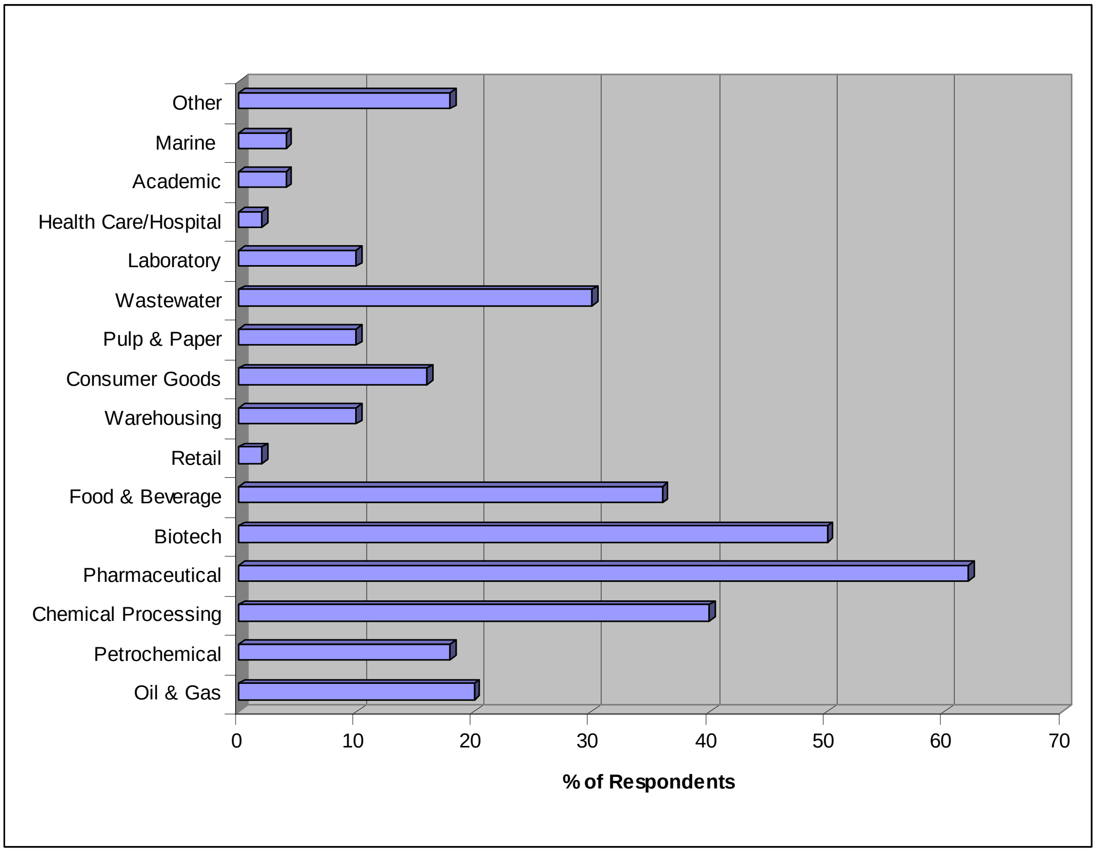

Figure 2. Major Industries Served by Respondents

A large percentage of the respondents (>30%) were from the ISA Certified Automation Professional (CAP) community. Due to the nature of this non-scientific survey[2], it is impossible with certainty to validate all responses. However, the authors' can attest that the Lessons Learned presented in this paper are from "real life", as they have first-hand knowledge and/or experience with each of the items presented.

When asked to select the median size of their typical systems integration project (services only, without hardware or software purchases), the respondents were almost evenly split between the following categories: $0 - $100k, $101k - $250k, $251k - $500k, $501k - $1MM, and >$1MM.

When asked to identify the major impediments to completing their scope of work on their last major integration project completed, the following was observed:

Problems encountered > 60% of the time:

- Engineering changes which resulted in rework/retesting
- Scope creep resulting in schedule pressure
- Scope creep resulting in budget pressure

Problems encountered between 40% and 60% of the time:

- Delay in receiving reviewed documents from the client
- Delay in receiving process data from the client
- Delay from client in making critical decisions

Problems encountered between 10 and 40% of the time:

- No synergy with the client
- Limited access to client or site
- Client slow in paying invoices
- Poor or difficult communications
- Poor scope definition
- Personnel turnover

These results indicate that the number one issue with systems integration projects is scope and engineering changes. Delays in the project are the second-most cited issue, with a handful of other issues occurring much less frequently. Given these major classes of observed issues, coupled with the top lessons learned described below, a clearer picture begins to emerge as to why some projects could have been executed better.

[2] A scientific survey would have required many additional demographic questions to be included in the survey, as well as identifying a more controlled sampling of systems integrators and end users. The goal of this paper is to present the views and experiences of systems integrators and end users in the context of presenting lessons learned from their experiences.

SYSTEMS INTEGRATORS: OUR LIST

Lesson #1: Don't Make Assumptions – If Unsure, Ask!

This is a very hard lesson to learn, and one that may be repeated over and over again. It is easy for brilliant systems integration resources such as ourselves to assume we know the answers to every question. Assumptions most of the time lead to incorrect decisions, which lead to rework, which, in turn, leads to addition costs, lost client goodwill, and schedule pressures. Question everything, including yourself. Repeat back to the client your understanding of the client's requirements to be absolutely sure all assumptions have been validated.

Lesson #2: Never Prepare a Cost Estimate Based on the Client's Estimate of Project Scope

How many times have you read a specification that includes phrases, such as "during the 8 weeks of software design" or "will only require a minor update to an existing phase"? These are projects that usually need to be "no bid". Project schedules and efforts required for specific tasks can be identified only after careful examination of the tasks involved and the risks that need to be overcome. Schedules, task lists, and efforts involved must be agreed to by mutual collaboration after a thorough understanding of the scope has been obtained.

Lesson #3: It Is OK to Say No

It is alright, and sometimes healthy, to say "NO" to a client. Repeat these lines until they flow naturally:

> "No, we don't have the resources to execute this project in a way to make the project successful."
>
> "We are going to no-bid this RFQ; there is too much risk involved to provide a lump-sum quotation. We will be glad to do it on a T&M basis."

Lesson #4: All Changes Have Cost and Schedule Impact

One of the most difficult lessons to learn is that there is no such thing as a change which as only schedule or just cost impact. The mere stopping to attend to the detailed analysis of a potential change has caused both, especially if other resources have been interrupted to help identify the impact of the change. Once the scope of a potential change has been broadcast, everyone working on the project has the details of that change in his or her mind, and it will affect the project execution. The remedy: build-in a "scope change allowance" into your project estimate, not to execute changes, but to identify a certain volume of change requests, knowing that it will take time and slow down the original effort.

Lesson #5: Staff Every Project With the Right Resources

This lesson is fraught with temptations. It is easy to assign the closest "free bodies" to a project to get them off overhead, but more harm than good can come from placing the wrong individuals on a project. At the most basic level, having the right people available starts with the company's hiring practices. New engineers or newly-hired engineers will require close supervision (plan for this additional time), and engineers participating outside of their normal areas of expertise add an additional risk to the project as well.

Lesson #6: Get Input From Operations about the Operations

This is also true for getting maintenance input from the maintenance department. Too often, systems integrators depend on the client project engineer or project manager to provide input, information, or direction without going to the source of the data. Unless your client contact is also in those capacities, you are not going to proceed with the best information. Do not be afraid to track down the operators, maintenance personnel, or instrument techs for input, suggestions, review comments, and preferences. How else might you realize colors on an HMI will be of no use to the many color-blind operators? How else might you know that small, drop-down menus on a touchscreen are a problem to a maintenance technician with a gloved hand? An operator will often not complain to a project manager about problems with the current control system: that is viewed as whining. However, telling the same to a systems integrator who is designing the new system. Well, that just might be salvation.

Lesson #7: Don’t Commit to a Lump Sum Proposal Unless the Scope is Well-Defined

Lump sum bidding is tricky enough, but when the scope is not well-defined, it becomes a total crap-shoot. Either the systems integrator will add so much contingency that they are priced off the job or they win the job and quickly find that the project is much larger than anticipated. In many cases, it is best to invoke Lesson #3 and just say “No bid”. Better yet, ask questions to help gain clarity on the details of the scope during the bidding process. From experience, it is often the case that the end user has no idea that their scope definition is filled with holes. Perhaps your insight and tenacity will not only help them improve the bidding process, but it may also give your company credibility as an experienced expert. Be careful, too much nit-picking and the client may regard you as a pest.

Lesson #8: Make Continuous Backups of Everything

This may seem to be quite obvious, but every systems integrator can tell you a horror story or two, if not a dozen. A systematic backup and restore procedure, a documented disaster recovery procedure, and a software and document traceability program should be in place for every project, no matter how large or small, prior to producing one line of code or one paragraph of a document. If you think an end user will give you a hassle about completing a task the first time, try telling them you have to do it all over again because a hard-drive failed or a power outage.

Lesson #9: Projects That Are “Just a Copy of” Aren’t

Many projects are billed as being “a copy of” an existing process or plant. And, we as systems integrators are much too willing to give the end user the benefit of the doubt, and our bids are structured on 90%+ reuse of code, graphics, etc. How many times has this one burned you? A good rule of thumb is that no project reuses more than 60% of the code, graphics, or reports from prior projects, processes, or installations. Advances in technology, increased process understanding, and lessons learned from the end users' perspective lead to the current project being similar to, but not just like the original. Besides, isn't that really the whole point to continuous improvement?

Lesson #10: Test Everything Alone, Then Together, Then Do It All Again

Do either of these scenarios sound familiar? The project schedule is in a crunch, so that last piece of code doesn't get tested until it is installed in the field, and it doesn't work. A software module used on a previous project is reused but is not tested, and during loop testing, it is found that scaling routine is incompatible with the new I/O subsystem. These are both real-life horror stories. From experience, the extra time and effort to perform an end-user witnessed FAT is worth more than the time taken to plan, execute, and document the testing. Module testing, integrated testing, and system testing has no substitute. You will perform these tests at some point in the project. Doing it for the first time in the field is not what you want to do. This is recipe for failure, increased expense, and client ill-will.

Lesson #11: Have a Change Management Plan In Place Before Signing the Contract

Projects always change. Sometimes, they change often. The quickest way to frustration with the work and with the end user is to not have a well-defined, agreed-upon change management plan prior to signing a contract to do the work. If the end-user does not provide a plan, have one ready to present. If the end user has a plan that is incomplete or not acceptable, don't be afraid to negotiate. Find a way to reduce paperwork on both sides, such as predetermined costs for adding an additional 20 analog I/O points to the project, inclusive of all documentation, configuration, code, and testing. This gives both the client and systems integrator a little leeway while reducing the paperwork burden for expected changes.

Lesson #12: Troubleshooting Is Not a Group Exercise

When a systems integrator is on site, all eyes are upon them. At some point in the project, usually during commissioning, the control system becomes the critical path. Extra scrutiny is placed on everything the systems integrator does. Learn to manage troubleshooting; it is as much an art as a science. Train your resources on how to detect, identify, and rectify problems. Teach them that troubleshooting is not at first a team sport or a group exercise. Use your eyes: check wiring, configurations, fuses, proper connections, power supplies, operation of instruments, etc. Eliminate the obvious and only then, call in for "backup". Most "group efforts" lead to circling around the same items already checked and searching under rocks on a "different path" altogether.

Lesson #13: Interfaces Are Planned; They Just Don't Happen

Whether the interface is with the end user or with an OEM supplier or equipment skid manufacturer, interfaces just don't happen. They must be planned, and planned well. For the interface with the end user, identify your single point of contact and make sure they know who yours is. For OEM suppliers, bypass the account manager or project manager and find the person responsible for the programming. You would be surprised how often this is a third party integrator. For equipment skid manufacturers, become a member of their review team so that there are no surprises later. Remember, however, their contract is not with you; it is with the end user. Find a way to let them know that the decisions they make affect your scope and effort. Share ideas and thoughts about data interfaces. Plan for this additional interface time and be cooperative. Nothing will shut down communications like a wayward attitude.

Lesson #14: No End User Has Ever Scolded an SI For Over-Communicating

Well, maybe a systems integrator has been told they communicate too much, but it would be an oddity. The authors have never been in trouble for providing concise, regular, honest, and complete reports and assessments to the end user on the status of the project. In contrast, the smoothest, best-run projects are the ones where this occurs, even if the end user has not requested the communications, reports, or data. The best rule of thumb: provide weekly reports with tasks completed, tasks to be completed in the next two weeks, identification of any issues, list of data needs, and an earned value or project cost analysis to the client whether required or not. You might like surprises, but end users hate them. Open, fluent, and frequent communication helps prevent surprises of all kinds.

END USERS: OUR LIST

Lesson #15: Treat Your SI as Your Partner – Communication

You hired your SI to design and implement a complex control system, not to be a mind reader. Openly communicate all thoughts, questions, and comments in a timely and well-documented manner so as to ensure the success of the SI effort; in a true partnership all partners succeed or fail together.

Lesson #16: Commit to Providing the Right Team Early On

There are few things more aggravating or costly in a project than “late blooming brilliance.” Commit early on to making available to the SI the people that can provide the right information in a timely manner and are vested with authority to make decisions. Representatives from design engineering, plant engineering, production, validation, and QA should be readily available on a regular basis to answer questions, review, and approve the efforts of the SI long before problems are experienced at FAT or at startup.

Lesson #17: Clearly Define the Expected Scope and Schedule with the SI

Clearly define the scope and required schedule of the project for your SI. For the best results contract documents such as P&IDs, formal Process Descriptions, Instrument Indexes, Data Sheets, I/O Lists, Control Panel Designs, and System Architecture should be issued in finished form. Key milestone dates such as system reviews, panel fabrication and testing, software simulation, FAT, commissioning, and startup should be clearly identified.

Lesson #18: Establish Clear Performance and Tracking Metrics

Come to an agreement early on with the SI on a methodology to accurately track and report their progress on the project. The tracking mechanism shouldn't be onerous and become a project in itself, just detailed enough to give everyone a good sense of where the SI stands on their progress. Progress on well defined SI deliverables such as Functional Specifications, graphic screens, alarms lists, control modules, and reports make for useful tracking parameters.

Lesson #19: Get Buy-In from End Users as the Project Progresses

Similar to #16 above, involve the end users of your system, the production operators and technicians, in the review and commenting process of the SI's work. Oftentimes, they will be better equipped and experienced to know how the final system should "look and feel" than the production supervisors and plant engineers; these people are a valuable resource: use them.

Lesson #20: Change Management – Managing your Internal Customers

Not all change is good. A haphazard and "afterthought" approach to providing comments and feeback to the SI is almost certain to result in change orders and negative schedule impacts on the SI's work. Clearly delineate "must haves" from "like to haves" and make reviewers commit, in writing, to their thoughts and comments. Any comment that is not documented does not exist. Period. All comments or requests that will result in changes to cost or schedule must be approved by the designated Project Manager and even then only after the SI has advised, in writing, of the cost and schedule impact of the proposed change. This is a common sense and disciplined approach to change management that is too often ignored in the heat of a project.

Lesson #21: Change Management – Managing your SI

As in #20 above, insist upon and enforce a firm rule that no changes are to be implemented regardless of who has requested them unless a formal submittal of cost and schedule change is made and approved.

Lesson #22: Review SI Submittals in a Timely Manner

Ensure that all SI submittals are reviewed and commented on in a timely manner. The best way to discover errors and misunderstandings early is to thoroughly review the submittals and to advise the SI of your comments well before they go "off in the weeds" or in a direction that is not in keeping with the site's desires. It is often difficult for the site team to carve time out of an already overstressed schedule to perform these reviews but it is an absolute necessity. The SI should be made aware that proceeding ahead on a particular portion of the design without an approved submittal may result in their "working at risk." It is mutually beneficial then for the SI and for the site to have all submittals reviewed in a timely manner.

Lesson #23: Leverage the SI's Work for Validation

Whenever possible, generate design, FAT, commissioning, and other official documents such as Functions Specifications in such a manner that they will satisfy Validation as well as project requirements. Get QA and Validation buy-in early on as to the format, content, and approval signatures for these documents that will allow them to be leveraged for formal Validation purposes at the end of the project.

Lesson #24: "Run Interference" for your SI

In addition to providing clear design documents, a dedicated, knowledgeable, and empowered internal project team, insightful and timely comments, and a disciplined approach to change management the best thing you can do for your SI is to "run interference." Assign a dedicated Project Manager who will, among other things, clear their path of obstacles and impediments that may be standing in their way.

The SI needs a face-to-face meeting with the production operators? Set it up. Where is a certain submittal in the review process? Find out for them. A meeting is needed to approve the format of the graphics screens? Gather the appropriate site staff in a review meeting with the SI. Quickly. Prompt attention to the requests and needs of the SI will result in a more cohesive and integrated effort. Everyone wins.

Lesson #25: Be an Active and Involved Partner in the SI Process – Take Ownership

It is difficult for a coach to direct his team from the locker room. The same goes for being "in the game" with your Systems Integrator. Being actively involved in the Systems Integration process is more than just answering questions. It involves raising questions, participating in review meetings, actively reviewing documents and software, measuring the end product against the requirements, and foreseeing and remedying potential conflicts and problems.

CONCLUSION

The work presented by the authors has given the reader a detailed look at the most common Lessons Learned in the Systems Integration industry. These lessons learned were each developed as the result of an experience, a hard knock, a sleepless night, and/or pure toil and sweat by systems integrators just like you. The hope of the authors is that these Lessons Learned will, as with world history, be a guide for future project work, with the intent of keeping the same mistakes from occurring again, helping to improve quality and efficiency of the systems integrator's project work, and providing a path to delight the client.

These are hard lessons to learn. Some will be repeated over and over again. The authors can attest to facing many of these lessons over and over again, project after project. Why? Because schedules will always be compressed and need to be met, scope will never be firm, allocations of funds will always be limited, and the requirements of the projects we lead will remain as fluid as the changing needs of the marketplace.

The best we can hope for is that we learn enough, and remember enough, and are strong enough to do our best with enthusiasm, integrity, and honesty. And that, perhaps, is the greatest Lesson Learned that the authors can share.

REFERENCES

1. Sommer, Scott, and Russell, Christopher, Systems Integration Lessons Learned Survey, conducted June 4, 2007, through August 3, 2007.