COPYRIGHT Notice

This paper was presented at CUMREC 2000, the Higher Education Administrative Technology conference. It is the intellectual property of the author(s). Permission to print out or disseminate all or part of this material is granted provided that the copies are not made or distributed for commercial advantage and that the title and authors of the paper appear. To copy or disseminate otherwise, or to republish in any form, requires written permission from the authors.
Implementation Projects: Land Mines and Other Lessons

John D. Wilder
Rensselaer Polytechnic Institute
Troy, NY

Abstract

Rensselaer Polytechnic Institute totally replaced its administrative computing systems in a series of projects during the 1990s. Along the way many lessons were learned, and recent projects have been consistently successful. This paper looks at the process by which new systems are implemented and highlights the lessons learned. Six phases are identified, covering the entire process from first concept and software selection through ongoing upgrade efforts and thirteen “Land Mines”, or common mistakes, are identified. Two large projects that implemented much of the Banner suite from Systems & Computer Technology Corporation (SCT) are used to provide examples.
This is a paper about system implementation projects, those long difficult journeys by which organizations move from an old set of technology to a new one. Following a trend that goes back to the 1980s these projects are far more common now than software development projects. Few organizations have the combination of size and uniqueness to justify developing software from scratch, and instead most will buy systems if they are available. Small systems, those with only a handful of end users and that are largely self-contained, are often installed with few problems. This paper looks at the implementation of large systems, those that span multiple business functions and may include a variety of technologies. These systems have assumed the central role in handling operations at many institutions.

Large implementation projects are long – usually spanning several years from first conception and investigation through the final wrap up, and they are complex, including a variety of technologies and crossing numerous organizational boundaries. Two other characteristics also contribute to their risk. First, these projects are often attempted by institutions after a long period of stability, implying that there is little organizational experience. Second, these projects can vary widely in nature as the technology evolves. Tales of woe are all too common.

In this paper I am going to concentrate on certain on “Land Mines”, mistakes that can rest beneath the surface of a project and then “blow up” to cause serious problems. To some people this may seem to be a rather dismal approach – concentrating on the negative, mistakes to be avoided, rather than on the positive. I don’t admit to being a pessimist, quite the opposite. Instead, I am concentrating on these points because all of these land mines have occurred on real projects that I have witnessed, both at Rensselaer and other institutions. It is more common for projects at different institutions to share these problems than the specific approaches to success, and in hindsight these mistakes are often easily visible. Many of these land mines are traps for optimists and project teams, at least the successful ones, are full of optimists.

It is only a coincidence that I list 13 land mines.

Rensselaer Polytechnic Institute moved through a typical cycle with its administrative systems over the last three decades. Early on in the 1960s and 1970s Rensselaer acquired computers and began to write software to support administrative processing. As the 1970s advanced, vendors appeared offering products that provided more and better functionality in comparison to what could be written in house. Some of this was acquired including portions of the administrative suite from Information Associates, Inc. This was installed with other administrative systems on an IBM mainframe running Michigan Terminal System, a state-of-the-art timesharing operating system that served a broad array of campus users.

After purchase the software was maintained in house without a great expenditure of effort. As a result the systems maintained the same level of late 1970s technology, which was noticeably out of date by 1989. In that year problems with the financial system prompted the university to begin investigating a replacement. Around the same time much of the academic computing began to move off of MTS and on to networked UNIX systems. The writing was on the wall for the future of the MTS system by the early 1990s. The project to replace the financial and payroll systems was only the first of many through the 1990s as, one by one, newer and better replacements were found for
Implementation Projects: Land Mines and Other Lessons

all of the systems supported by MTS. This process was only completed this year, 1999, with the final shutdown of MTS.

The first project, called FAIMS (Financial and Administrative Information Management System) implemented the Banner Finance and Human Resources system from Systems & Computer Technology Corporation (SCT), with software selection occurring in 1990 and the implementation running from 1991 through 1994. The project was a success, but only eventually and with a great deal of pain and disruption. Starting in 1994 the university convened another group to define a replacement for all of the student services systems. This group also selected the SCT Banner product, the Student and Financial Aid Systems, and in contrast to FAIMS this product was installed during 1996-1999 with great success.

What has become apparent from these and numerous smaller projects is that implementations tend to move through phases. The idea of phases is not radical per se: it is reasonable to expect that similar efforts would pass through similar evolutions. However, dividing an effort into phases helps highlight the differences in the activities at each stage. What’s more, it becomes more apparent how mistakes made in one stage in turn affect activities in later ones. Mistakes are often cumulative, with problems in the early phases persisting through the course of the entire project, and the effects multiply in combination with other mistakes.

In this paper I discuss six phases that cover the entire life of a project. These phases and their characteristics are listed in Figure 1.

Phase 1: Foundation

The foundation phase of the project is the beginning, the time when nothing may be known but the need for something new. During this phase the project takes form until, at the end of the phase, it is approved and implementation work can start. In many ways the Foundation phase is the most important because it is during this time that the end goal is defined along with the means to get there. Like a building, it is not unusual to find problems appearing in other parts that can be traced back to problems in the Foundation.

The end result of the activity during the first phase is a project definition, whether written or tacitly understood. This definition answers crucial questions including:

- What overall results are expected as part of this project? How will improvements be visible to students, faculty, staff, alumni, or other stakeholders?
- What technical solution is required?
- What resources (time, money, and people) are required?
- What is the high level plan for implementing the solution? What timeframe can be expected and how will the project be organized?

Land Mine 1: Assuming that choosing a system is the same is defining the project.
At Rensselaer a Selection Committee was established in July 1994 to select software to replace the aging student and financial aid systems. Even at this early stage the committee received some funds so that committee members could take trips to other schools, attend conferences and, if needed, consultants could be engaged. The charter for this committee included writing the project definition as described above in addition to recommending a specific package.

Over the next fifteen months the committee carried out a variety of activities, including:

- In conjunction with university leadership several objectives and guiding principles were defined. These included important decisions such as favoring an integrated solution rather than a series of “best of breed” systems, aiming for “state of the industry” rather than “state of the art”, and implementing with the understanding that there would be “minimal modifications to the baseline software”.
- Mapping business processes. This activity quickly provided invaluable information on required functionality and the maps were important in developing a formal Request For Information (RFI).
- Visiting several other schools to see a variety of systems in daily use.
- Inviting several vendors to provide demonstrations before writing the RFI.
- Issuing the RFI, reviewing and comparing the responses, and then inviting vendors to provide detailed demonstrations based on Rensselaer developed scenarios.
- Engaging a respected consulting firm to review the selection process and facilitate two key decision-making meetings.
- Developing a high-level project plan including a schedule of milestones, hardware and software requirements, staffing needs, and budget.
- Benchmarking the project definition against similar projects either underway or complete at a number of other schools.

These activities resulted in a solid software selection, SCT Banner, and a complete project definition. In addition the process addressed several other people issues that are important to the success of any project. First, the group was able to achieve consensus on the correct solution. This did not happen overnight. It took time and a chance for everyone to work together before trust could be established among committee members, and more time for issues to be raised and addressed. A comprehensive process allowed the group time and the mechanisms to deal with conflicting aims and perceptions. The second issue addressed was the ability of the group to get work done. Early on the expectation was set that the project would be long and time-intensive for all involved. Over time a culture developed within the committee that this was real work (as opposed to many other committees) and that everyone would need to work hard and contribute.

The third issue dealt with the expectations for the project by the campus community, especially the university leadership. Earlier projects had often suffered from “the squeeze”, a phenomena in which the people selling the project play up the benefits while the budget request is successively trimmed by repeated layers of review. The obvious result is a project that is oversold and underfunded. In this case the project proposal avoided playing up the opportunities for miraculous improvement. The comprehensive planning process allowed a firm budget of 5.4 million dollars to be proposed (including
$250,000 of unallocated contingency) and maintained through the entire approval process.

**Land Mine 2: “The Squeeze”. Overselling and underfunding the project in an effort to secure approval.**

In the end the project developed a firm foundation. A solution and a plan to implement it were defined and the resources were secured to make the project a success. Everyone had a clear expectation with regard to the amount of work and risk involved along with realistic expectations about the benefits that would be achieved.

The carefully crafted project plan then suffered the ultimate disaster: it was approved.

**Phase 2: Getting Started**

The phase “Getting Started” covers the period during which work begins on the implementation up to the time that a smoothly functioning project is in place. Much of the effort during this time is simply spent in securing resources and getting things moving. Just because a project has been approved does not mean that work can begin the next day. Organizations need time to adjust while priorities are changed and prior commitments honored. In fact, one of the most common problems is that this stage is simply left out of the plan. Assumptions are made such as “in week two the vendor will be onsite to install a test system”, ignoring the reality that there is no on-site staff available yet to work with the vendor. Even more extreme are cases where complicated contracts must be negotiated or that staff or consultants must be brought in from the outside, each of which can take weeks. Ignoring these activities can lead to projects that are weeks behind before they even get started.

**Land Mine 3: “The land that time forgot”. Not allowing time to get the project started and the team in place.**

The Banner project at Rensselaer had allowed for startup when the plan was put together, but it did confront another uncomfortable reality. Implementation projects have schedules that are tied to the annual business cycle. For example, a new admissions system would typically be implemented to support the recruiting or application processing for a certain term. A financial system will most often go live with the start of a new fiscal year. For the Banner project the target was to bring functions up in time for a fall term, with a schedule of go-lives starting 10 months after approval. This plan was immediately squeezed before it started. As is often the case, the approval process moved to its own rhythms. At Rensselaer, the approval came two and one half months later than had been expected. Since dates such as the start of a semester are not
flexible the project team was left with the choice of either making up the time or shifting the schedule out almost a year. The decision was made to move ahead with the shortened schedule.

As any project gets moving there are a many activities. As shown in Figure 2, one way to view these is to separate them into the project startup tasks versus the tasks of implementation. On the one hand the project startup tasks concern aspects of the entire project, while the initial implementation tasks concentrate on the near term, dealing with crucial items that are on the critical path. Figure 2 provides examples of each.

This phase calls for balance – indeed this is the original high-wire balancing act. Too much emphasis on project startup leads to paralysis as all of the effort is consumed on planning and preparation. Too much emphasis on the near-term implementation tasks leads to near term progress and long term confusion. A good rule of thumb is to focus near-term activities on the tasks necessary to bring up a test installation that can be used for training. This is usually on the critical path because of the lead time involved in procuring and installing the necessary hardware and software, and the requirement that this system be in place prior to functional training.

Land Mine 4: “The schizophrenia of startup”. Not dealing with both long term planning AND the near term issues of getting a project in motion.

Working towards the short term goal of a training system highlights a series of necessary dependencies, namely:

1. Making business decisions on how to configure the system requires that people be knowledgeable (trained) on the system and have an installation available for experimentation.

2. To conduct functional training requires a system to train on so that people can learn “hands on”.

3. A training system requires hardware and software.

4. Hardware and software require that technical staff be available and skilled with the technology.

5. For technical staff to be skilled may require training on new technology.

Leaving out one or more of these steps is all too common.

Land Mine 5: “Breaking the chain”. Technical training > skilled support staff > training system > product training > business decision making.
Phase 3: Stealth

The third phase is the “stealth” phase, the time when the project, now approved and moving forward, falls off the radar screen of the institution and becomes a concern only to those who are immediately involved. It is during this “in between” time that most of the work is accomplished. Indeed, this is the last time period to address issues before going live, the last opportunity to ensure success.

The first problem many projects face is that old friend to most human endeavors, procrastination. Large projects are cursed with long timelines. In many cases the schedule may call for more than a year between startup and the first “go-live”. Human nature being what it is, anything that is not happening within the next few weeks exists in a world of the indefinite future, a world that can always be worried about the next day. Everyone has enough crises to dominate “now” without worrying about projects that still have months to go. Obviously the problem with this picture is that there are months of work to do in the months allotted: the extra time is an illusion. While everyone usually accepts the idea that the project will have everyone overloaded come go-live, the challenge is to induce a sense of urgency from the beginning while there is still time to do something.

One approach is to simply talk up the problem in concrete terms. Using shorter units of time can help this. A message that “such and such must be ready in 27 weeks” can focus more attention than saying it must be ready sometime in April next year. Similarly, calendars that list the number of working days left before certain milestones helps state the time in an immediate form. Saying that something must be ready in 43 days can have more impact than “two months”.

A more definitive approach is to set intermediate milestones, tangible achievements that help everyone focus in the short run. Typical milestones can include the readiness of the training system, the completion of vendor training, or the first round completion of the business decision-making and the system configuration.

Land Mine 6: “I love you, tomorrow…” Procrastination.

Another common trap is to lose sight of the solution while everyone concentrates on getting the system operational and fully configured. There’s no doubt that there’s much work to do: technology must be installed, the system must be configured with codes and rules, and who can forget reports. In this case it is easy for the system to seduce because it is now concrete. Checklists appear which list all of the tables that must be loaded, lists of existing and proposed reports are drafted, and data conversions mapped out. Slowly, the process of bringing the system to life begins to drive the project.

The danger with this is that the system by itself is not the solution. Many less tangible activities are also crucial, for example, ensuring that solutions are defined for the gaps in the software, developing a communication and training plan that covers the campus rollout, drafting and finalizing new policies and procedures. The complete solution covers the entire business process, and the system should not be allowed to eclipse that.
A similar problem arises as people begin to focus on specific parts of the system. Large and complex systems are difficult to deal with as a whole so the work must be split up into pieces. One group worries about generating bills, another about registering students, while a third focuses on admissions. In turn each group concentrates on its specific area of responsibility, configuring and testing to ensure each function works properly. What this view can miss is the overall process itself. Can a student be applied, accepted, registered for classes, and then billed? Does the process flow smoothly from one stage to the next? As assumptions or changes are made in one area, are they being reflected in the others?

_Land Mine 7: Letting the system block the view of the solution.
Land Mine 8: Assuming that the whole = sum of the parts. Maybe it is, but maybe it isn’t._

These types of problems are easily compounded by faith that all of the new technology and the system itself will work properly on the first try, and it will work exactly as expected. It rarely does. The best training course in the world is no substitute for hands on experience that works the system as it will be used at the institution. And there is nothing like exercising the system with all of the idiosyncrasies of the local implementation to find a unique bug that no other customer has ever seen.

Not adhering to this in the early stage of the Banner project was the single largest mistake. In this case the problem was not even with the new modules, but with products being upgraded in preparation for the additional installation. Early signs of potential problems were overlooked. There was no reason to expect that there were serious problems with some specific versions of the underlying Oracle tools. Instead, the upgrade work progressed slowly as problems were worked out one-by-one. By the time the situation came to a head the project had lost several months and required a complete reworking of the schedule. This was preferable to the disaster that would have happened had the project pressed forward and brought the upgrades live. However, more extensive testing early on would have uncovered the problems and allowed them to be dealt with up front, not in crisis several months down the road.

_Land Mine 9: Faith that the system will work correctly and as expected._

Later during the Banner project the decision was made that the university should move straight to web-based registration, eliminating the existing paper-based process entirely. This implied a new process, using new functions based on the new technology. The high profile/high risk nature of the decision made the need for testing obvious, and a skeptical eye was turned on all aspects of the process. Testing was accomplished on several levels:

1. Initial product testing to ensure that the web system worked properly and as expected. Several defects were found and corrected along with several areas in which the product was found to behave differently than expected.
2. The web product required extensive configuration, including the development of instructions and help text for students. This was developed and then tested by having about forty students register on a test system. The students were observed during the process and provided feedback afterward. This allowed another series of problems to be found and improvements made.

3. The technology was then tested to ensure that it could scale and handle the full load of registration and not just a handful of testers. Automated tools were used to simulate large numbers of web users and a number of problems were found and corrected.

4. The entire process was “beta” tested a few days before the live registration by allowing about sixty students to register early.

In the end registration came off with remarkably few problems. What is more important is that each form of testing uncovered problems that would have involved real students and faculty had they not been caught and corrected. This was not unusual – every testing process throughout the entire project brought to light problems that were then corrected.

Only testing can uncover problems in time, and only process testing, testing that covers an entire life of a student, employee, and so forth, can ensure that solution is driving the system and that the parts will add up to the whole.

Doing all of the above is a tremendous amount of work. For the people working on the project many other responsibilities are pushed aside so tasks can be completed and checked off. It can then come as a shock to be walking across the campus and, in casual conversation with someone not involved with the project, be asked “so whatever happened to that project? Is anything going on?”. It is all too easy for busy members of a project team to forget that for most people on the campus the project was an article in the school paper and that it has since disappeared. Nothing has changed yet so nothing is visible. This is, after all, the stealth phase. But it shouldn’t be, at least not entirely.

The challenge for the project team is to communicate what other people in the community should be aware of before “go live”. With earlier projects at Rensselaer the tendency had been to ignore this, with the result that when changes began to happen many people were caught off guard. Communication, when it happened, tended to focus on the specific items or details, not on the overall project process. With the Banner implementation a different tact was taken and the project steering committee explicitly assumed the task of providing information to the campus community. A communication plan was drawn up outlining what information would be communicated through what means, and when it should happen. People outside the project team were kept up to speed on the progress and the general nature and timing of upcoming changes.

*Land Mine 10: “So is anything going on…”* Not communicating to people outside the project team what is happening and what they can expect.
Phase 4: Going Live

Going live is the high point of the project. Expectations are high as new things begin to happen and the process of cutting over from the old systems to the new begins. Months of preparation begin to pay off as the new system starts working for real. Ideally months of careful planning and coordination have the system and the process for cutting over perfectly positioned for success.

The reality is not as pretty: “going live” is a crisis and it calls for crisis management. This is different from the project management that moves the project through the earlier stages. Events begin to happen quickly and what counts is the ability to respond. This too must be planned: the tools must be in place ahead of time to manage the crisis. Communication paths and problem reporting mechanisms must be worked out in advance. Most important is a tracking system (even on paper) for handling incoming problem reports. Various approaches have been used at Rensselaer, from spreadsheets to homebuilt software and purchased packages. During the last major system upgrade approximately 200 problem reports were generated during testing and the go-live process. Tracking software allowed each one to be captured and then tracked as additional information was gathered and the problem resolved.

Land Mine 11: Believing that “going live” won’t be a crisis.

Then there is the day after the system is live. “Going live” is the activity that draws the most attention and planning effort. It is very easy to go too far with this and not spend enough time planning for what happens through the rest of the business cycle. Most of the attention and testing tend to focus on what will be needed immediately after the system is live, ignoring later but equally crucial needs. Again, a whole process perspective is important to help avoid this. What is important is not only that a student can be registered, but that the add/drop process will work smoothly, class lists can be generated, and enrollment reports produced. Downstream the bills will need to be printed and eventually instructors will need to grade. It is only natural that the highly visible initial events will get attention, but problems in any part of the process will rapidly become visible in the worst way.

Land Mine 12: Not planning for the day (week, month...) after the system goes live.

Phase 5: It Isn’t Over Yet

Going live is not the end, it is only another beginning. We move from the stage of “going live” to the stage of “firsts”, namely, the first time various functions are carried out on the new system. Using the previous example, a student information system might go live with scheduling and registration, but that is then followed by many firsts; add/drop, class lists, enrollment reports, and so on. Each of these requires something new, some additional effort, and they are all bounded by deadlines imposed by the academic
calendar. These in turn happen while other functions are still going live, e.g. billing or housing.

This stage can be difficult for the support staff to juggle as project demands run up against the crisis of the day. In this stage support demands are cumulative, at least for a considerable time. It is typical for two full business cycles (semesters, academic years, fiscal years) to go by until support demands drop to a lower steady state. Not that there isn’t any fluctuation, but it normally takes going through a process twice to get all of the pieces in place and working properly. The first time through short cuts taken and it isn’t until the second time through that all of the kinks are worked out.

The Banner Finance/HR implementation at Rensselaer ran into this problem. The system was phased in with three “go lives” spaced over a year. Each additional function that went live then drew off implementation staff into ongoing support. This was acceptable for the first release (Purchasing and A/P), but after the second release (Payroll/HR) there weren’t enough people left to support the third release (G/L, Research Accounting, Budget). The system went live, but work was deferred, leading to cumbersome manual workarounds that only added more work. Resources were focused on fixing operational problems at the expense of other non-critical but important items such as departmental reporting. The result was a large population of unhappy people across the campus and a backlog of work that took eighteen months to dig out from.

**Land Mine 13: Not having the people to deal with problems on the “live” system while others are still being implemented.**

**Phase 6: Upgrades**

Once a new system is in place life does not go back to the way that it was. Especially when moving from system that was supported in-house to one that is vendor supported. The good news is that the vendor is continually improving the product and working on new releases. That is also the bad news, at least for people who want to forget about the implementation. The implementation never ends. Each major release is a scaled down version of the initial implementation, calling for attention and involvement from all of the same people. Almost as soon as one completes the planning starts up for the next.

At Rensselaer we have addressed this by largely keeping the project organization intact. It has evolved and expanded, but it has not gone away even as the level of effort has dropped. The hardest part is not forgetting the lessons learned.

**So What Does It Mean?**

Just a few final observations. Most of the land mines occur early on, with the first five covering events before the project even gets moving and another six applying to activities that go on before anything goes live. This should not be surprising since the
land mines refer to the problems with the project itself, not the system being implemented. It is during the early phases that the project itself is defined and conversely, is most susceptible to problems. Beyond that, once a problem occurs it will not go away by itself. It won’t be solved without deliberate and sustained efforts to correct it.

The good news is that the land mines are avoidable. None of them are problems so obscure that they can’t been seen and avoided. The key is making the effort to look and avoid. While I am sure that there are other land mines out there, avoiding these will move any project a long ways towards a successful completion.
Figures and Tables

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foundation</td>
<td>The project is defined and high level decisions made about the solution and the resources required to implement.</td>
</tr>
<tr>
<td>Getting Started</td>
<td>The project starts, people are organized, and more detailed plans are made.</td>
</tr>
<tr>
<td>Stealth</td>
<td>The project has high levels of activity but is not visible to people not immediately involved.</td>
</tr>
<tr>
<td>Going Live</td>
<td>Operations begin moving over to the new system.</td>
</tr>
<tr>
<td>It isn’t over yet</td>
<td>The project continues bringing functions live until every business event has happened at least once.</td>
</tr>
<tr>
<td>Upgrades</td>
<td>The project is done but the work continues on an evolutionary basis.</td>
</tr>
</tbody>
</table>

*Figure 1. Six Project Phases*

<table>
<thead>
<tr>
<th>Project start-up tasks</th>
<th>Implementation Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building the project organization – committees, work groups etc.</td>
<td>Finalizing contract</td>
</tr>
<tr>
<td>Staffing the project</td>
<td>Procuring hardware and software</td>
</tr>
<tr>
<td>Adding detail to project plans</td>
<td>Installing a system for testing &amp; training</td>
</tr>
<tr>
<td>Technical training</td>
<td>Analyzing business processes and starting change</td>
</tr>
<tr>
<td>Long term planning</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 2. Project start-up tasks versus implementation tasks.*
Land Mine Summary

Land Mine 1: Assuming that choosing a system is the same is defining the project.

Land Mine 2: “The Squeeze”. Overselling and underfunding the project in an effort to secure approval.

Land Mine 3: “The land that time forgot”. Not allowing time to get the project started and the team in place.

Land Mine 4: “The schizophrenia of startup”. Not dealing with both long term planning AND the short term issues of getting a project in motion.

Land Mine 5: “Breaking the chain”. Technical training > skilled support staff > training system > product training > business decision making.

Land Mine 6: “I love you, tomorrow…” Procrastination.

Land Mine 7: Letting the system block the view of the solution.

Land Mine 8: Assuming that the whole = sum of the parts. Maybe it is, but maybe it isn't.

Land Mine 9: Faith that the system will work correctly and as expected.

Land Mine 10: “So is anything going on…” Not communicating to people outside the project team what is happening and what they can expect.

Land Mine 11: Believing that “going live” won’t be a crisis.

Land Mine 12: Not planning for the day (week, month…) after the system goes live.

Land Mine 13: Not having the people to deal with problems on the “live” system while others are still being implemented.