Payoff

Lack of management attention to a common set of factors has been shown to contribute to the failure of large-scale outsourced systems development projects. IS managers who follow the recommendations presented in this article will help ensure on-time and within-budget delivery of effective large-scale systems.

Problems Addressed

The outsourcing of large-scale systems development activities is becoming more prevalent, yet the success of these activities is not. In many cases, the repeated failure of large-scale outsourcing projects is caused by lack of management attention to a common set of factors. Given the magnitude of investment involved in large-scale efforts, IS managers must learn to manage these factors before a system is approved for funding and in a way that ensures on-time and within-budget delivery of effective systems. This article draws from the experience of several large-scale systems development efforts in the public sector to present recommendations that help ensure the success of large-scale outsourced development projects.

Challenges In Large-Scale Projects

Well-known failures in the area of large-scale systems development include the State of California Department of Motor Vehicles (DMV) registration system, the Denver Airport baggage-handling system, and the Bank of America trust system. Failure, however, is not reserved to the well known. A study of 24 large US companies reports that 68% of major projects using client/server technology overshot their schedules, 55% overspent their budgets, and 88% involved significant redesign. Large projects (i.e., those in excess of 5,000 function points) are canceled about a quarter of the time, and projects in the 10,000 to 20,000 function point range are canceled half the time. About 67% of projects that are completed overshoot their original schedules and budgets by nearly 100%.

IS managers responsible for projects that are significantly larger than others they have managed face challenges not only regarding day-to-day management and integration but also in terms of their ability to conceptualize whether the scale factor will affect system operation. Missed intermediate milestones in large projects involving new technology require special attention, because the development is occurring in relatively uncharted territory. It is at these milestones that managers need to make decisions to continue the project, cut back on functionality, or spend additional money. If such decisions are postponed, the sponsors of the system may find themselves at a point of no return.

Factors to Manage In Large-Scale Projects

Analysis of outsourced systems development projects ranging in value from $1 million to $100 million and involving, among others, the State of California, County of Los Angeles, and Los Angeles Unified School District revealed several factors that significantly

---

contributed to failure or delay in multiple instances. In each project, managers failed to give appropriate consideration to one or more of the following areas:

- Acceptance testing standards.
- Contract specifications, especially:
  - Measurable business effects or outcomes.
  - Requirements specifications.
  - Arbitration and cost-reimbursement clauses.
  - Mechanisms for settling management disputes.


Because absence of proper attention to these factors contributes to failed systems development efforts, management of these factors is a vital, although not necessarily sufficient, ingredient for ensuring project success.

**Developing Appropriate Acceptance Testing Standards**

Standards for acceptance testing should be adopted for all systems development efforts. IS managers should ensure that any deviation from such standards is approved by an independent technical body and that wherever possible a parallel acceptance test is conducted.

Parallel testing need not be conducted in real time, which can reduce costs, but any live or parallel test must be sufficiently long to ensure that conditions occurring at significant calendar times throughout the year are experienced. There must be no cold turkey starts of a system without adequate testing of critical system capabilities, and there must always be a fallback plan in case of failure.

IS managers should ensure that proper documentation of the acceptance test is maintained and that ad-hoc judgment is not allowed to substitute for contractually obligated performance. When the new and the old systems can accept the same transactions, comparison is facilitated. When they cannot, a set of specialized transactions needs to be devised to fit the new system.

Under such circumstances, it is possible to encounter a reconciliation-of-reports problem between the new and old systems. IS managers should not let the issue rest with a promise from developers that these minor errors will be fixed down the road. Independent sign-off that the acceptance test plan meets standards and that the test actually done corresponds to the planned test helps ensure that the area of acceptance testing does not become the cause of a failed development effort.

**Ensuring Clear and Comprehensive Contract Specifications**

IS managers must ensure that technical contract specifications are clear and based on design objectives. Before a request for proposal (RFP) is issued, the IS manager should hire independent technical and legal professionals to review and approve the specifications.
Such a review should focus not only on what is included but also on possible areas of omission (e.g., lack of a binding-arbitration clause). The contract should clearly delineate resources to be supplied by the vendor and the user in terms of number of person hours, qualification of personnel, calendar time when people are to be available, hardware, documentation, and training. The manager must also ensure that all contracts contain specifications to cover various contingencies.

Independent review of contract specifications clearly has costs in terms of money and time. Is the insurance worth buying? Although managers have to make that judgment for themselves, it is a well-known fact that ambiguous contract specifications lead to lawsuits.

In Los Angeles County, independent review by outside legal counsel following review by in-house counsel has produced valuable results. In addition, independent technical review outside of the sponsoring agency is being instituted through a new office of the chief information officer. Previously, independent departments or users issued their own RFPs without central oversight.

**Specifying Measurable Business Effects or Outcomes**

IS managers should ensure that measurable outcomes or expectations from the implementation of a system are included in contracts at the outset, preferably as stipulations but at a minimum as goals. Examples of such outcomes include a reduction in the amount of manual labor required and increases in the efficiency of service (e.g., the amount of paperwork needing to be redone will be reduced from 2% to 1%). Failure to meet such specifications would be cause for cancellation of the project and for arbitration damages.

IS managers should ensure that the amount of manual intervention and support required by any new system once operational is also detailed. Lack of this specificity can lead to delivery of all kinds of fancy systems that do not achieve business outcomes.

Consider the example of a new order-entry system. High-level business outcomes for the system could include the following:

- Reductions in staff that save * million dollars a year.
- Reduction of one minute in the average length of time a customer spends on the phone while the clerk fills in the screen.

These expected outcomes might be difficult to include in a contract, however, because a vendor might balk at being responsible for total user information performance and the actions of user personnel. Still, an effort to include such outcomes should be made.

**Specifying Business and System Requirements in Clear and Cost-Effective Terms**

IS managers must ensure that contract specifications clearly reflect the user's business requirements. It should not be assumed that a given system will magically meet specific user requirements even though it may contain laudatory general capabilities. The full requirements of the user must be anticipated for several years, and systems acquired must be flexible enough to accommodate such requirements.

System requirements should be based on the system's handling most, but not necessarily all, transactions. Automated handling of 100% of transactions may necessitate the development of excessive code and generate additional development expense.
Ensuring that a stated requirement accurately reflects the needs of the user can be accomplished through independent evaluation of business and systems requirements. Some RFPs have been found to contain 20 to 30 pages of minute requirements, many of which the user did not need, and to omit others that the user did need. If an outside agency is hired to evaluate requirements, it should be made liable for the validity of its findings.

**Including Binding-Arbitration and Cost-Reimbursement Clauses**

IS managers must make every effort to ensure the inclusion of a contractual requirement for binding arbitration to handle disputes and a penalty structure agreed upon by both sides. Penalties, such as treble damages for revenue losses resulting from lack of system performance, should be immediately payable without further legal recourse. Los Angeles County, for example, lost a substantial amount of money from inaccurate Medicare and Medicaid billing resulting from information produced by an outsourced system that turned out to be faulty.

Contracts should stipulate that a vendor guarantee financial resources to cover the cost of a fallback position that permits the continuation of the client's essential user services. Alternatively, contract provisions should ensure that the user is reimbursed for losses incurred as a result of curtailment of essential services. The size of a performance bond, if any, should be commensurate with the size of the project and the dollar magnitude of the operation it concerns.

Is it realistic to try to include binding-arbitration, cost-reimbursement, or measurable-effects clauses into contracts? Although it has been difficult to include such clauses into public contracts in the past, progress is being made in this area. Smaller vendors, however, are still hesitant or incapable of providing these type of guarantees.

**Developing Dispute-Settlement, Change-Control, and Early-Warning Mechanisms**

Because any project involving a user and a developer will have disputes and changes, IS managers must ensure that appropriate mechanisms are in place to meet such contingencies as they arise. Basic management practices and alertness warrant that such mechanisms be enforced as well; a sound structure that is not enforced serves no purpose.

Any systems development contract should therefore include a variety of levels for settling disputes, starting with gentle persuasion and ending with CEO summits. The failure of lower-level management to settle a matter or escalate it to a higher level of review should be a cause for disciplinary action. Failure to meet milestones or provide deliverables should be pursued within the managerial chains of both contractor and user.

IS managers should ensure that the outsourcing contract contains an adequate mechanism for change control that is fair to both the user and contractor. Early-warning conditions must be specified, and IS staff must be kept cognizant of development or implementation progress. Potential problems should be brought to the attention of appropriate managers.

**Developing Effective Project Management**

Project management is a major factor in the success or failure of systems development efforts. It involves tracking progress and then making changes to keep projects in control and heading along the path to completion in an efficient manner. Projects should be
managed by an experienced project manager who has been involved in the development of systems of the same dollar magnitude. When an individual with this experience is not available internally, IS managers should consider hiring someone to fill this function.

Effective project management requires establishing a standard--including well-defined project review cycles--that considers deliverables, effectiveness of what is delivered, metrics (e.g., function points), and tightly defined reporting formats. Progress tracking can only be accomplished when there is a well-defined and unambiguous measure to track. Function point analysis meets this criterion and is suitable as well for change control and measurement of scope creep (i.e., percent change in function points between phases of the software development life cycle). The methodology is also readily communicated to users.

IS managers should recognize that use of any metric can involve the fudging of numbers at lower levels. This is particularly commonplace with percentage of completion estimates. A savvy manager, however, should be able to detect or have a feel for how things are going. If they are going badly, the manager should know how easy or difficult the cure or fix is going to be.

Some managers, however, simply do not want to face up to the fact that a project is in trouble. Although these managers know, either overtly or subconsciously, when a project is in trouble, the seemingly dire consequences of shutdown or other drastic remedial action lead them to decide not to acknowledge reality and just hope for the best. Some of the factors that play a role in such cases follow:

· Top management, for its own political agenda, has made successful completion of the system a matter of paramount importance.

· IS managers find continued operation under the present system such a distasteful prospect that even a slim chance at producing a better system is deemed more desirable than an aborted effort or rescoping of functionality.

· Managers have developed an affection for the new system that hinders them from acknowledging the tell-tale signs of failure.

· The reward/punishment system favors going forward.

Disregarding the facts, for whatever reasons, can cause IS managers to continue a development effort that should be curtailed or altered. Once certain levels of commission are passed, it becomes increasingly difficult to stop a project. The manager gets more deeply enmeshed and hangs on to increasingly unrealistic hopes for success.

**Allocating Sufficient Budget Resources For Implementation**

The success of outsourced systems development also depends on the sufficient budgeting of resources for system implementation, including user training, dual operation, conversion, physical facility changes, and documentation. To accommodate peak personnel requirements during implementation and changeover, IS managers should consider the use of temporary employees and overtime and allocate funds for these purposes.

Training funds should be budgeted not only for the people doing the training but also for the people taking the training. These individuals require overtime or replacement help to do their jobs while they are being trained. IS managers should advise user managers in advance of the need for such allocations.
Because a thorough evaluation of implementation costs increases the price of a system, some IS managers do not conduct full cost evaluations. In addition, overruns in development or programming costs lead managers to co-mingle implementation funds with operating funds or even with other funds for development of the system.

The practice of robbing Peter to pay Paul or banking on the idea that there always will be a way to find the money for implementation is fraught with danger. It usually results in inadequate or shoddy implementation. The saddest of all development experiences is to see the hard work of systems development go to waste or at a minimum used inadequately because of improper implementation. The following sample scenario illustrates the results of this practice.

A lack of implementation funds for a newly development welfare-payments system causes users to be improperly trained. As a result, a number of transactions are mishandled. Needy members of the public do not receive their allotted benefits. The media picks up on this, and the system is accused of having all sorts of glitches. Governmental committees hold hearings, and consultants and advisory committees are called in to investigate what was in fact an effectively developed but poorly implemented system.

The preceding scenario could have been avoided had the IS manager set aside adequate funds for implementation and safeguarded the funds from other use. An independent assessment of the adequacy of implementation resources also helps to avoid problems later on.

Examining Software

In any turnkey system, provision must be made for examination of the source code. This examination provides an assessment of how easily user-required changes can be accommodated, as well as the ease with which the user can assume control of the system if necessary. Examination of the source code can be done by user personnel or by an agreed-upon third party and should address the documentation and structure of the code, methodology used in development, ease of change, modularity, and programming language.

Shortening the Length of the Development Cycle

The length of time between the inception of system specification and the end of system implementation must be kept to a minimum. IS managers should endeavor to cut major systems development projects into smaller more manageable projects that have shorter lead times and are more amenable to careful oversight.

The rapid pace of changing requirements and technology mandate that the development cycle never be allowed to exceed two years. Projects with estimated completion time exceeding two years should be redefined; IS managers should consider using ready-made software and integrating if possible. Although this approach might seem inefficient, the nature of today's development environment warrants its use.

Requiring Independent Quality Assurance

Large-scale outsourced systems development efforts benefit from independently monitored quality assurance. Hiring an independent entity represents an additional expense, but it provides an insurance policy for management that is justified by the high stakes of large projects. Los Angeles County is using this approach to ensure that when a user department
outsources a large-scale project, an independent contractor monitors vendor quality and adherence to standards.

**Ensuring Technology Transfer**

IS managers need to ensure that appropriate technology transfer to the user agency or department occurs. This transfer involves the careful delineation of the information that must be transferred to the sponsoring agency and of the form in which the information will be transferred. Managers who leave these issues up to contractor personnel are courting disaster, particularly in the area of system maintenance.

**Manifesting System Ownership**

Because having a high-level manager take a personal interest in a system (i.e., manifesting ownership) helps keep everybody involved in the development on their toes and ensure that things get done, personal attention by user and IS managers at all levels is certainly another key ingredient of a successful outsourced development effort.

How can an IS manager tell that system ownership has occurred among colleagues? The amount of time other managers spend on the project and the number of questions they ask regarding the system are some of the indicators of system ownership. But most important is an attitude that demonstrates involvement.

**Conclusion**

The need to specify requirements and objectives has been a prime consideration throughout this approach. Sometimes, however, such specificity is not available, as is the case when a project involves changing conditions or the development objectives become determined as the project progresses. Such conditions are not ideal for outsourcing, except perhaps on a time and materials basis, and reinforce the need for careful consideration of the outsourcing decision.

Once such a decision has been made, IS managers should actively manage the factors known to have a direct effect on the success or failure of development efforts. Before a system is approved for funding, a manager must secure detailed presentations from the contractor describing how these factors have been addressed. Furthermore, periodic checks should be made during the systems development life cycle to ensure that management of these factors has been kept on track.

Does consideration of all the factors ensure a project's success? Not necessarily, but it does improve the project's chances of success. IS managers who follow the recommendations provided in this article help ensure that if outsourced systems development efforts fail in the future, it should at least be for reasons other than the ones described here.

**Author Biographies**

Raoul J. Freeman

Raoul J. Freeman is professor and chair of computer information systems at California State University-Dominguez Hills and chairman of the Data Processing and Telecommunications Advisory Committee of Los Angeles County. He has extensive experience in both the public and private sectors in large-scale systems implementation,