

Outsourced projects exhibit many of the same characteristics as in-house projects—and some significant differences. How can you successfully outsource a project without losing control?

K E Y S T O S U C C E S S F U L S O F T W A R E P R O J E C T O U T S O U R C I N G

KEYS TO SUCCESSFUL SOFTWARE PROJECT OUTSOURCING

Software project outsourcing is the practice of paying an outside organization to develop a software application instead of developing it in-house. Outsourcing provides benefits including improved company focus, accelerated development times, and access to expert software development capabilities. Project outcome depends on a few critical success factors: specifying requirements in detail; making an explicit decision to outsource; obtaining adequate resources; selecting a qualified vendor; writing a contract; and building an in-house management team. If these steps are followed, outsourced projects can be runaway successes. When not, projects fail at an alarming rate. Lessons learned from outsourced projects apply to in-house projects as well.

Contents

A Lesson From the Trenches	2
Keys to Successful Outsourcing	4

Additional materials on:

Summary of Keys to Success	3
Decision to Outsource	5
Two-Phase Acquisition	7
Outsourcing Resources	13

Software project outsourcing allows companies to turn over critical projects to companies with specialized software capabilities. Outsourcing vendors can make qualified staff available sooner, leading to more timely project completion. Outsourcing can greatly reduce the amount of in-house effort devoted to a project, allowing a client company to focus on its core competencies while software specialists focus on the software.

Outsourced projects exhibit many of the same characteristics as in-house projects—and some significant differences. How can you successfully outsource a project without losing control?

A Lesson From the Trenches

Giga-Corp was the market leader in its vertical market software category. Giga-Corp's customers had been clamoring for months for the second version of its popular *GraphIt* software product.

The product concept for Version 2 was for an incremental upgrade using the same technology that had been used on the previous version. Giga-Corp's key programmers were experts in the product's business area, but they would be tied up on another project for 3-5 months. The manager in charge of the project was overwhelmed with other projects, so he decided to outsource.

Giga-Corp created a 50-page requirements specification and sent a Request for Proposal (RFP) to a dozen vendors. The RFP didn't include a budget, but it did indicate a strong preference for a fixed price bid and delivery within five months. Giga-Corp indicated that price would be the most significant criterion. Giga-Corp held a bidder's conference to

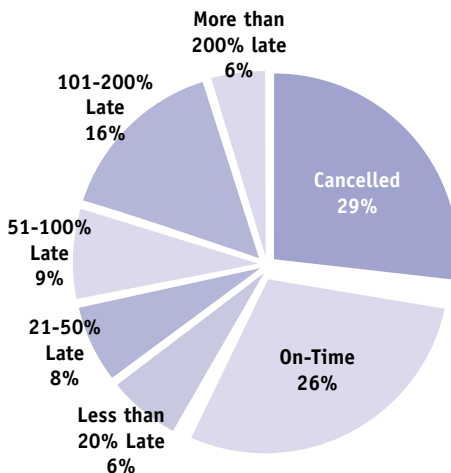


Figure 1
A surprisingly high percentage of projects—both in house and outsourced—are delivered late or cancelled. Only about one quarter of all projects are delivered on time. Source: "Turning Chaos into Success." *Software Magazine*, December 1999.

answer vendor's questions, and about a dozen vendors attended. The day of the proposal deadline, Giga-Corp received four bids:

Tech-Co, a software projects company, submitted a time and materials bid of \$1.2 million with an 11-month schedule.

Fledgling-Co, a startup company founded by one of the senior programmers who had worked on Version 1.0 of *GraphIt*, submitted a time and materials bid of \$900,000 with a 12-month schedule.

Mega-Staff, the project arm of another software body shop, submitted a time and materials bid of \$400,000 with a 5-month schedule.

People-R-Us, a programming body shop without any experience in whole-project development, submitted a time and materials bid of \$300,000, with a 5-month schedule.

Giga-Corp threw out the high and the low bids, reasoning that those were outliers and probably the result of misunderstanding the project's technical requirements. It liked Fledgling Co's experience with its specific project, but it felt that its management abilities were unproved. It selected Mega-Staff's bid based on the depth of staff that Mega-Staff was able to apply to the project and then negotiated a reduction in Mega-Staff's estimate to \$350,000.

Heading South

The first indisputable hint of trouble came when Mega-Staff's head recruiter phoned Fledgling Co's president on a Thursday afternoon, sounding desperate. "We need three C++ programmers by Monday," he said. "We submitted a time and materials bid based on implementation in Visual Basic. We threw in a C++ bid to hedge our bets, but Giga-Corp said

Summary of Keys to Success

Managers of successful outsourced projects observe several keys to success:

- Understand company goals and objectives; is outsourcing the right solution?
- Create an outsourcing plan.
- Define the software requirements in detail.
- Create a Request for Proposal (RFP).
- Estimate your project's cost and schedule before finalizing your RFP.
- Obtain sufficient budget and management resources to assure success.
- Select a qualified vendor.
- If no vendor appears qualified, switch to a two-phase acquisition model or bring the project in-house.
- Create the outsourcing contract with care, and have it reviewed.
- Actively manage the outsourced project.
- Use outside experts when needed.

it would choose a vendor based on cost. We priced the C++ option \$50,000 higher, so we were sure they would take our VB bid. We won the project, but Giga-Corp's senior engineer said our Visual Basic approach wasn't technically feasible. We don't have enough C++ programmers to support the project." Fledgling Co's president said he couldn't find three C++ programmers on such short notice and wished Mega-Staff's recruiter good luck.

Somehow, when Monday rolled around, Mega-Staff showed up for the kickoff meeting at Giga-Corp headquarters with three new C++ programmers. The Mega-Staff team was enthusiastic and appeared to make good progress at first, but by the five-month mark, the team was nowhere close to completion. Mega-Staff rescheduled and promised delivery within a total of eight months, which was acceptable to Giga-Corp because that coincided with the beginning of its next sales year.

Eight months into the project, the end was still far from reach, and Giga-Corp demanded cost concessions for late delivery. After that, Mega-Staff announced several more 6 week schedule slips.

Fourteen months into the project, after more schedule slips and many painful meetings with Giga-Corp, Mega-Staff delivered about 25 percent of *GraphIt's* planned functionality at a cost of approximately \$1.5 million. Considering the combined effect of the cost and schedule overruns and the functionality underrun, in real terms Mega-Staff had overrun its proposed cost by 1,400 percent and its proposed schedule by 1,000 percent.

Customers immediately complained about the software's low quality, and within a few weeks Giga-Corp and Mega-Staff had to release a maintenance update.

Keys to Successful Outsourcing

With software outsourcing on the rise, the Giga-Corp and Mega-Staff disaster is being replayed by software outsourcing clients and vendors throughout the world. These problems can be avoided if the participants understand a few software outsourcing fundamentals:

1. Specify requirements in detail
 2. Make an explicit decision to outsource
 3. Obtain resources
 4. Select a vendor
 5. Write the contract
 6. Build an in-house management team to monitor the project to completion.
-

Step One: Specify Requirements

The most common causes of project failure are incomplete, inaccurate, and unstable requirements. Project success depends on painstaking requirements management whether a project is conducted in house or outsourced.

Because the statement of requirements often becomes part of the legal contract between the client and vendor, an outsourced project has an especially strong need for high-quality requirements. If requirements aren't clearly specified, the project can become a battleground on which all possible ramifications of the client's and seller's different interests are contested.

As the case study illustrated, this is a battle that both parties will lose. If the organization doing the buying doesn't have the in-house resources to create detailed requirements, it should use a two-phase acquisition. During the first phase, the client outsources

development of the requirements. During the second phase, the client puts those requirements out for bid and then outsources the development of the software. In general, if an organization doesn't have the technical or managerial resources in house to perform the necessary tasks for effective outsourcing, it should have an expert third party perform them.

Step Two: Decide to Outsource

The decision to outsource invariably involves weighing both costs and benefits. Because outsourcing can increase a project's risk, you should make the decision explicitly, with full awareness of potential problems. **See "Decision to Outsource" above for more details.**

Step Three: Obtain Resources

The client must acquire sufficient resources to complete the project. These resources include budgets for the proposal phase, vendor payment, and management of the project on the client's side.

Decision to Outsource

Here are 10 common reasons that companies outsource software projects.

1. Improve company focus.
2. Make use of vendor's expertise in leading-edge technologies.
3. Free internal resources for other purposes.
4. Achieve development schedules not possible with internal staff.
5. Reduce and control operating costs.
6. Gain access to world-class capabilities.
7. Leverage vendor's specialized project management experience.
8. Scale up for a specific project without increasing number of permanent staff.
9. Avoid historic problems with internal projects that have been difficult to manage or out of control.
10. Share project risk with another company.

Estimating Resources

The detailed requirements created in Step One form the basis of the client's effort and schedule estimates. A client who doesn't create detailed estimates or have them prepared by a third party won't know whether a \$300,000 or a \$1,200,000 bid is more realistic. As the case study illustrated most vendors tend to underestimate. Basing your budget on vendor estimates is a sure-fire recipe for budget and schedule overruns.

If the client lacks the in-house expertise to create accurate cost and schedule estimates, it should have them created by a third party.

Contract Options

You have several ways to structure the contract for an outsourced project. Each has its own strengths and weaknesses.

With a *Time and Materials* arrangement, the client pays the vendor for each hour worked. The vendor builds its profit into its hourly billing rate. The client accepts the risk of possible cost overrun. This arrangement minimizes risk for the vendor and is appropriate when requirements cannot be precisely defined at the beginning of the project. It is also appropriate when a client wants to maintain maximum flexibility to meet a specific schedule, functionality, or budget target.

With a *Cost Plus* contract, the vendor charges its internal labor rate plus a fixed amount of profit, for example, an average billing rate of \$100/hour plus \$250,000 fixed profit. If the project costs more than the vendor has estimated, the client pays only for the vendor's additional costs, not for additional profit.

Variations on cost plus contracts include *Cost Plus Incentives* (adding incentive payments for early delivery, high quality, or cost underruns) and *Cost Plus Penalties* (adding penalties for late delivery, low quality, or cost overruns).

In *Fixed Price* contracts, the vendor agrees to deliver specific functionality for a price specified in advance. Fixed price contracts are appropriate when the project requirements are very well defined and stable. In theory, this arrangement minimizes risks for the client.

An innovative variation on the standard fixed price arrangement is *Two-Phase Fixed Price* in which the vendor performs early design work under one fixed price contract, then performs implementation work under a follow-on fixed price contract. This arrangement significantly reduces risk for the vendor, while providing the client with more cost control than is possible with cost-based contracts. **See the sidebar "Two-Phase Acquisition" on next page.**

Two-Phase Acquisition

Many projects can benefit from being broken into two phases. In two-phase acquisition, requirements development and planning work is conducted during the first phase, and detailed development work is conducted during the second phase.

Two-phase acquisition gives the clients more control over vendor selection. If the client is satisfied with the work performed by the vendor during Phase 1, the client may continue with the same vendor during Phase 2. If not, the client can use the materials generated during Phase 1 to put the project out for bid again, and hire a different vendor for Phase 2.

Generally, Phase 1 will consume 10-20 percent of the resources and 15-30 percent of the calendar time on a project. The activities performed during Phase 1 tend to be more open ended than those performed during Phase 2, which accounts for the variability.

A client should use two-phase acquisition when:

- Software requirements are not well defined or seem to be constantly changing.
- The clients wants a formal checkpoint at which it can assess whether to continue with the rest of the project.
- The client wishes to share risk with the vendor. Breaking the project into two fixed price phases is a more equitable sharing of risk than time and materials contracts (in which the client bears all the risk) and fixed price contracts (in which the vendor bears all the risk).

Successful fixed price projects depend on defining requirements carefully at the beginning of the project and then tracking requirements changes—and their impacts on cost and schedule—throughout the remainder of the project.

Step Four: Select a Vendor

The choice of vendor is the most important choice a client makes on an outsourced project. A well-run vendor selection process usually entails creating an RFP, distributing it to potential vendors, and systematically evaluating vendor proposals.

Create an RFI

A solid RFP is critical to the success of an outsourced project, so a sophisticated client solicits help from its trusted vendors before finalizing an RFP. During this Request for Information (RFI) stage, vendors review a draft RFP and provide input. Vendors naturally encourage the client to create an RFP that will be favorable to their specific capabilities, but they also point out general weaknesses in the RFP before it's used as the basis for a multi-million dollar project. The RFI process helps to improve the quality of information the client ultimately obtains through the proposal process.

Create the RFP

The RFP should contain at least the following materials:

Software Requirements Specification. These are the detailed requirements that were developed in Step One.

Statement of Work. This is a document containing the management requirements for the project. It specifies your demands regarding managing changes to requirements, the kinds of tools and development environments that will be used, technical methods and methodologies, software documentation, engineering data (design notes, diagrams, scaffolding code, build scripts, and so on), backup procedures, source code control, quality assurance procedures, and the project's management organization—especially the interaction between your organization and the vendor.

Documentation list. The RFP should specify the documents you want to have developed in conjunction with the software. You should include design documents, source code documentation, operating manuals, status reports, and any other documents you want developed.

Cost and schedule estimates. Include the cost and schedule estimates you prepared during requirements development. Giga-Corp didn't know how much its project should cost or how long it should take, which led it to hire a vendor incapable of performing the work. If you don't publish your cost and schedule estimates in the RFP, the proposal process degenerates into a game in which each vendor tries to guess your budget.

In the case study, the divergence among the vendor's cost estimates suggested that either the technical requirements were not developed well enough or that some or all of the vendors were poor estimators. The fact that the highest bid was submitted by the company most familiar with the software was a clear warning. If Giga-Corp didn't want to choose the highest bidder, it would have been wise to switch to a two-phase acquisition approach,

using one of the vendors to develop the requirements more fully, creating new cost and schedule estimates, and then putting the project out to bid again.

Problems can take control of a project if not addressed early enough in the project. Be sure to address problems quickly, when they are easy to correct.

Evaluation criteria. Tell the vendors what criteria you will use to evaluate their proposals. Typical criteria include project management capability, general technical capability, technical design approach, technical methodologies, technical documentation, engineering data management, requirements management approach, configuration management approach, and quality assurance approach.

Competent vendors will explain in their proposals how they plan to meet each of the evaluation criteria. By publishing evaluation criteria, you make it easier for yourself to make side-by-side comparisons among proposals, and you significantly improve the quality of information that vendors provide.

Proposal preparation guidelines. Describe how you want the proposals to be organized and formatted. Include descriptions and page limits for each section, margin sizes, font sizes, and so on. Consider creating a sample proposal so that you know the page count limits are reasonable. This might seem overly fussy, but specifying a standard proposal format makes the job of proposal evaluation easier and pays off in the long run.

The RFP process requires a lot of work. Relying on an established relationship with a trusted vendor can be a viable alternative, especially on smaller projects.

SW-CMM

One general means of evaluating vendors is by using the Software Engineering Institute's Capability Maturity Model for Software (SW-CMM). The SW-CMM is currently the most credible credential for software organizational effectiveness. On the SW-CMM's five-point scale, most organizations are at Level 1, which represents a limited level of effectiveness **(including limited estimation effectiveness, as shown in Figure 2 on next page)**. The most effective organizations operate at Levels 3–5, which present dramatic improvements in cost, schedule, and quality performance. A Level 3 organization typically produces software in 60 percent of the time, with one third of the effort, and about 10 percent of the defects of a Level 1 organization.

Good project management practices provide high leverage influence over project success regardless of the specific technologies employed. A winning tennis pro might have a

higher-tech tennis racket than an amateur, but difference in the two players' ability is not primarily a result of the technology. If the pro and the amateur switch rackets, the pro will still beat the amateur. The same is true with software development. Technology matters, but technique matters much more.

Pick the Winner

Careful RFP creation makes proposal evaluation quick and effective. Create a decision matrix based on the evaluation criteria described in your RFP, and score each proposal accordingly. Be prepared to follow up with questions to cover missing or insufficient proposal responses.

You might eliminate some vendors because of low scores overall or in specific categories. For example, you might eliminate any vendor that scores less than 7 out of 10 points in project management capability. By publishing explicit evaluation criteria and scoring proposals based on them, Giga-Corp could have known that Mega-Staff was critically weak in project management. Thus, Mega-Staff's eventual project overrun would have been predictable—and avoidable—at proposal time.

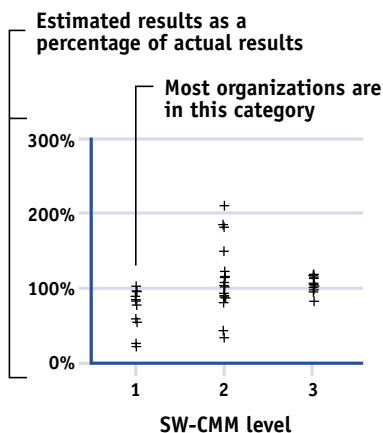


Figure 2
Software project estimation ranges from poor to excellent. Most organizations tend to severely underestimate their projects. Source: *A Correlational Study of the CMM and Software Development Performance*, Crosstalk, September 1995.

As the Giga-Corp case illustrates, the lowest bid and the final project price are often not related. The goal of awarding a contract is not to choose the lowest bid but to choose the vendor that will provide the best value—do the best job for the least money as judged by systematic evaluations of the proposals. Even though Tech-Co and Mega-Staff submitted similarly priced bids, Tech-Co scored significantly higher. Giga-Corp might have decided that Tech-Co represented a better value for the money.

Remember that the evaluation matrix scores represent soft data—they are numeric, but they are based on subjective judgments.

You can also use the evaluation criteria for negotiating changes to the winning vendor's proposal. For instance, Giga-Corp could have

Example of a Proposal Evaluation Matrix

Criteria	Points Possible	People-R-Us	Tech-Co.	Mega-Staff	Fledgling-Co.
Project management capability	10	3	8	5	3
Application-area experience	10	1	5	3	10
Technical-area experience	10	8	10	3	10
General technical capability	10	5	10	10	7
Organizational strength	10	5	9	10	3
Technical design approach	10	5	8	8	10
Technical methodologies	10	5	8	7	8
Requirements management approach	10	4	10	7	9
Technical documentation	5	3	5	4	5
Engineering data management	5	1	5	4	5
Configuration management approach	5	2	5	3	5
Quality assurance approach	5	1	5	4	4
Total points	100	43	88	68	79
Overall ranking	-	4	1	3	2

negotiated changes to requirements management and quality assurance approaches if those were the weak points in the winning bid.

Step Five: Write the Contract

Your warm feelings about your choice of vendor will quickly evaporate if the project begins to unravel. Be sure your contract spells out the details of management requirements, technical requirements, warranties, patents and other intellectual property issues, contract termination, payment, and any other important issues. Be sure your contract means what you think it means. Spend the money to have the contract created (or at least reviewed) by legal counsel before you sign it.

Step Six: Monitor the Project

The magnitude of Mega-Staff's overrun suggests that Giga-Corp essentially didn't monitor or control the project after the contract was awarded. Giga-Corp's managers might have voluntarily accepted a 100-percent or 200-percent cost overrun, but they would have to be trying to put themselves out of business to knowingly accept an overrun of 1,400 percent.

The most common mistake in managing outsourced software development is that no one on the client side manages the outsourced development project at all. While outsourcing can indeed reduce the amount of management needed, it increases the degree of management *sophistication* needed. The problems involved with managing a project across the hall are magnified when you have to manage a project across town or across the globe.

Because you can't monitor progress on an outsourced project just by walking around and talking to project team members, project tracking indicators must be more formal. In your RFP, you should specify what management controls you want the vendor to provide so you can track the project's progress. Such controls might include:

- Weekly status reports
- Weekly reports on the requirements specified and requirements that have been fully tested
- Weekly updates to the list of technical risks
- Weekly defect statistics
- Weekly reports on the number of modules planned, number designed, number coded, and number that have passed their reviews
- Monthly status review meetings

On small projects, someone from the client's side must be put in charge of monitoring the vendor's progress and working to resolve any problems that arise. On large projects, several people from the client's side might be needed.

Smooth Sailing

These steps might seem like a lot of work for an outsourced project. As the case study illustrated, however, the price of not doing your homework can negate the benefits of outsourcing.

Outsourcing can save you time, effort, and money, and it can allow your staff to focus on other critical projects. The time you spend laying the groundwork for a successful project and then steering the project to the desired conclusion is time well spent. And many of the practices that make outsourced projects work will help your in-house projects succeed too.

Outsourcing Resources

Software project success is often determined within the first 10–20 percent of the project. Even though the vendor you select will do the majority of the project work, your actions during the early parts of an outsourced project will have a significant influence on its success. Here are some resources that can help.

Construx Software (construx.com) Construx Software offers consulting services for RFP creation and evaluation, requirements workshops, third-party estimation, project planning, and vendor selection. Construx also provides full custom software solutions.

Software Project Survival Guide website (construx.com/survivalguide/)

The website related to Steve McConnell's best selling *Software Project Survival Guide* contains many resources for ensuring software project success of both in-house and outsourced projects.

Outsourcing Institute (outsourcing.com) This site contains a variety of outsourcing-related resources including guides to vendor selection, critical success factors, and outsourcing discussion groups.

Project Management Institute (pmi.org) PMI provides general guidelines for project management including certification and training.

Software Engineering Institute (www.sei.cmu.edu) This site contains a description of the Capability Maturity Model for Software (SW-CMM), the most widely recognized credential for software vendors.

For more information about Construx Software's services, please visit our website at www.construx.com.

For more information about Construx Software's services,
please visit our website at www.construx.com

Construx[®]
delivering software project success