



# **How to Mitigate Risks Related to Software Project Outsourcing**

**John Hsieh  
Easy! Software LLC  
[www.ez-sw.com](http://www.ez-sw.com)**

## **How to Mitigate Risks Related to Software Project Outsourcing**

Nowadays many U.S. firms are outsourcing their software projects to India, China and Eastern European countries to save money. In 2007, a CIO Insight survey of 265 senior IT decision-makers, 47% noted that they were under more pressure to reduce costs compared to the prior year. Given the recent global economic downturn, it shouldn't be a surprise to see the percentage continue to rise in an accelerated rate.

IT systems often are the lifeblood of an organization. As such, IT outsourcing should not only be a strategic decision, but also be accompanied with a well thought-out execution plan.

IT outsourcing can mean to outsource your work such as customer help desk operation, IT infrastructure maintenance and management, software project implementation, etc. This paper will focus on the software project outsourcing and explore many critical factors, which will determine the success or failure of a project. Understanding these factors will allow the project manager (or whoever in the organization responsible for the outsourced project) to put a plan in place to ensure a successful outcome. After all, it is your business, which relies on the system, and your career, which to a certain degree depends on the success of the system that you oversee.

People normally define project success as delivering a complete system within budget, on schedule and with expected quality. Outsourcing project development to a less expensive offshore vendor is a popular option to reduce the costs, but this doesn't mean it is worry free; it still require close management or else disastrous outcome may and can happen. Another way of defining success is to extend the measure throughout the system lifecycle, which includes on-going maintenance, software upgrades, new feature, new functionality, etc. We will use the latter broader definition for our discussion.

### **Schedule**

Proactively monitor a project progress is no doubt a better way than passively reading vendor's progress report. It is not uncommon that vendors use ad-hoc lifecycle management approach to manage their projects out of saving money and time, but at the cost of eventual system quality. This rudimentary approach creates unreliable measures. Additionally, human factor can also play into an artificial sunny-day progress report -- this is particularly common for the percentage completion payment agreement. The project manager shall insist that the vendor put in place an automatic tool so that project status can be generated unbiased and accessible by the project manager with ease and in real-time instead of unscientifically collected and generated data. Only with a reliable project status can the project manager assess the overall project progress and take necessary actions before it is too late.

### **Project Quality**

The development cost saving shall be from the labor rate difference rather than from the offshore vendors doing less or taking shortcut; either one will no doubt has negative impact on system quality.

If you are a believer of prevention is better than correction, then there ought to be a well defined process and best practices in place with enforcement mechanism to ensure that errors are not introduced to the system in the first place.

One way to specify a project quality is to spell it out in an outsource contract such as acceptance criteria. However, no matter how detail an acceptance test plan is written and how thorough the test is executed, it is not possible to identify all defects during the acceptance test. Besides, quality is best achieved by continuous monitoring and improvement rather than just measure it at an end point.

To avoid the last-minute surprise, the project manager should have a reliable means to constantly monitor the quality and to immediately take necessary corrective actions. For example, if a feature is complete and defects are detected during the feature testing phase, the project manager should be aware of them and make sure they are fixed as soon as possible rather than letting other new feature development sidetracks them and later on they become the culprits of poor quality and schedule delay.

In some industries you may have regulatory quality guidelines to follow. Under this situation, the project manager needs to make sure the vendor fully compliant with these guidelines and auditable records are available.

### **Updates and Maintenance**

One reality of software system is that it will never be defect free. As such, bug fixes shall be expected. Additionally, very often you will have the need to either enhance the existing functionality or add new ones to your system. In either case, this type of need will more likely arise some time after the initial system is delivered. The question, are the original developers still working with the vendor and, if they are, are they available? An extreme scenario, but may not be uncommon, is that most of those knowledgeable developers are gone and there are no documents (requirement, design, change records, source code comments, etc.) available for new developers to pick up from.

Obviously, it's not reasonable to expect a vendor to keep its developers around for this reason. However, you should be able to and must require the vendor to produce the documents mentioned earlier and keep them up to date so that new developers can quickly be up to speed for the new work. The added costs of long learning curve and trial-and-error required to bring up new developers are often overlooked, but in reality they should be a part of the total cost.

### **Conclusion**

Besides economical reason, firms may outsource their software project because it is not their core business. Their employees don't have the technical expertise required to implement the system in house. However, there ought to be someone, who is knowledgeable and experienced, assigned to oversee the development activity. In addition, it is imperative that you have the fallback plans to protect your company from the offshore outsourcing failures. This may mean to go back to your old system or roll back to your old business process.

The Easy!Flow from Easy! Software ([www.ez-sw.com](http://www.ez-sw.com)) is a complete project and process management solution, which addresses all the issues discussed in this document and more. Its various functions are completely integrated so that all team members share and update one single source of project information rather than scattered, incomplete and often contradictory information from non-integrated systems. It facilitates collaboration and communications among team members. Though packed with many functions and features, it is designed with ease to learn and use in mind to minimize the overhead yet maximize the benefits of using it. For instance, each individual member can customize personal preferences, queries, reports, etc. that are relevant to his/her role in the team. Take the role of the project manager, mentioned so many times in this document, as an example; s/he can easily create reports that reveal relevant information pertaining to the overall project quality and schedule. Easy!Flow offers many predefined PM metrics, queries and reports, from which you can choose and pin them to the UI shortcuts to make everyone's life easier and more productive. It even offers out-of-the-box development processes for your team to jump start your project, whether it is a stringent process or an Agile one. Though with a very rich set of features, Easy!Flow is very affordable compared to other less complete products. The combinations of low overhead, low price and great benefits make it an ideal application lifecycle management system for any outsourcing vendors as well as companies that outsource their projects.

### **About the Author**

Mr. John J. Hsieh has more than 20 years of experience in software development. He worked at Bell Labs for ten years where he was involved in development of many large-scale and mission-critical systems. He cofounded CommTech Corp., which developed software systems for telecom applications. In CommTech he led a software group with more than 120 software engineers. He is the founder of Easy! Software LLC ([www.ez-sw.com](http://www.ez-sw.com)). Mr. Hsieh has BS and MS degrees in electronic engineering, and a master degree in business administration. For any comments, you can reach him at: [jjh888@easysoftwarellc.com](mailto:jjh888@easysoftwarellc.com).