

8 Ways to Overcome Your Fear of Outsourcing

Authored by: **Administrator** [help@knowledgebase-script.com]

Saved From: <http://www.articlediary.com/article/8-ways-to-overcome-your-fear-of-outsourcing-65.html>

You have probably heard the stories: Software that does not work. Software that does the wrong thing. Money paid to set up an offshore operation that never produces any software. Can these outsourcing nightmares be avoided? Of course! The key is how you manage the **outsourcing** of your **software development**.

The two biggest fears of outsourcing that I hear from clients are the fear of selecting a bad outsourcing vendor and the fear of not managing their work effectively. Let's assume you pick a great team, perhaps one of the proven Accelerance teams from around the world. Even then you may still have lingering worries about getting your software developed quickly. Here are eight ways you can manage your outsourcing to get reliable results.

1. Milestone tracking - Your plan for outsourcing should include some estimating and scheduling. The outsourced team should commit to dates for the completion of the milestones. Define your development milestones so they occur frequently and make corrections if they are not being met.
2. Requirements tracking - Make sure the features you need are being completed properly in each release. You want to make sure your users will be able to gain the maximum benefit from your software. You don't want important details to be lost or forgotten.
3. Daily monitoring of your outsourcing progress - No news is NOT good news. You need regular communication with your outsourced team. This should occur almost everyday. You should at least have a regular email or a conference call on specific days to make sure things are going well and that unresolved issues are not blocking progress.
4. Continuous builds - Put a system in place to re-compile and build your software every time a developer checks in a source file. This allows you to detect problems as they occur so they can be fixed quickly and do not lead to a delayed software release.
5. Test-driven development - You should require your outsourced engineers to perform unit tests of their programming work. Tests should be created first, before the feature is implemented. Writing the test first helps clarify what the software should do. It also builds a set of regression tests that can be performed later to check for basic system functionality.
6. Integration testing and Quality Assurance - Testing of the individual software modules by the outsourced developers is not enough. You also need to test the entire system to make sure high levels of functionality and quality will be experienced by your users. Comprehensive testing requires automation to check all paths through the user interface. A full QA (Quality Assurance) process also verifies the software behaves well when users make "mistakes."
7. Collaboration for design and problem resolution - If the engineers are working in your office then you just meet with them to resolve the issue face-to-face. Of course, this is not possible when the engineers are thousands of miles away across one or more oceans. But with the Internet you can get pretty darn close. An email or instant message conversation can often resolve your issues. Using the telephone is also an extremely useful tool and with VOIP software like Skype, Yahoo Messenger or Google Talk the communication cost is zero. The important point is that the tools are available to make long-distance collaboration a reality and they enable extensive use of outsourcing to be practical.
8. Knowledge management and retention - You provide a specification to the outsourced team. Issues are resolved and clarifications are made. Then the software is delivered and it works well. But what if the outsourced team leaves the picture? Will you be able to pick up the software and make changes efficiently? Yes, if you have gained the knowledge of how the software is written. The technical documentation describing the architecture and workings of your software should be created by you or your outsourced team. You don't need a huge tome no one will read. Simple installation instructions and architectural documents that can facilitate the learning process and enable your US engineers to

deploy and support your software are sufficient.