
An offshore development war story

29 July 2005

John Ferguson Smart

Abstract

Offshore development is not simple : it raises technical, organizational and managerial issues which require solid project management and a good dose of innovative thinking. This article is a case study of one such project : what worked, what didn't, and what corrections were applied. Places and names have been changed.

Introduction

Imagine a new offshore factory in an oriental country. A sizable mission-critical project decomposed into 2 phases has been assigned to the offshore team. It is a J2EE project relying heavily on Open Source technologies : Postgres, JDK 1.5, Hibernate, Struts, JSTL, JSP 2 and JasperReports, running on Tomcat 5.5. The offshore development team is in the process of being recruited and trained.

Critical times

Our story starts around the time the phase 1 development activities were scheduled to be getting up to speed. At this point, project monitoring has started giving alarm signals :

- Delays in milestone deliveries were becoming apparent
- Code reviews showed poor coding quality, disparate coding style, beginner errors, and dubious architectural choices.
- Preliminary alpha testing is giving very poor results : many test cases cannot even be completed.
- The project management reserve was dwindling daily.

Faced with this increasingly alarming situation, Jack, the project manager, studies the situation with Mohamed, the offshore team leader, in one of their regular meetings using IM or IP-telephone. During this meeting and the ones that followed, several major issues are implicated:

- A high developer turnover, resulting in an under-sized team and a lot of ongoing training. The team of 6 had just lost 2 trained developers to a major multi-national IT firm. Another had left the previous month. The scheduled team size at this point in the project was 8, with 5 Java developers, whereas in reality they were reduced to 5, with just 2 Java developers.
- Management preferred to look for already-qualified and locally-trained engineers, rather than to provide costly tailored on-site training. However, Open-Source Java skills were harder to find on the job market than initially thought. As a result, the offshore team leader was experiencing difficulties getting the development team up to strength and operational.
- Developers who were found were often lacking in experience: many had trouble applying architectural recommendations and best practices, creating an overhead in code reviews and corrections.

The situation was effectively critical : delays were slipping fast and the code that was written was badly written and buggy. Action had to be taken, and fast!

Initial solutions

Jack decided to attack the problem from several angles :

- Bring in local reinforcements to work on the project
- Tailor offshore recruitment techniques to the local job market
- Provide individual coaching where needed to raise the level of the offshore team

Local Reinforcements

A team of 3 experienced local developers were added to the project. They were based in the European HQ. They had never met the offshore team. They were able to implement an important module in the Phase 1 part of the project, allowing the offshore team to concentrate on the rest. There were however some

organizational downsides to this solution : Cultural differences, political issues, and some cases of individual negative attitudes, created friction between the two teams.

Tailoring offshore recruitment techniques

One thing Jack and Mohamed had noticed about the offshore development team was this : one of the best developers had no previous Java experience, whereas some very mediocre ones had many years experience. However this developer was smart and curious, and eager to learn.

From this point one, less emphasis was placed on precise IT skill-sets, and the recruitment process privileged looking for smart, curious people capable of learning quickly. A trial period ensured that this fairly intuitive approach could be verified.

Individual training and coaching

This recruitment technique naturally required additional effort in individual training and coaching. Although time consuming, this helped foster a team development culture, and allowed a first-hand experience of the developer's aptitude for the job.

Slowly, a stable core team of competent programmers was established on the offshore site. Code quality, best practiced and test-driven development approaches are fostered and encouraged, and are progressively adopted by the whole team. Though these recommendations date from the very beginning of the project, their application cannot be effective in practice without a stable and well-trained development team.

Process optimizations

In Phase 2, many of the lessons learnt in phase 1 were applied to optimize the development process :

- In Phase 2, for scheduling reasons, the local reinforcements were retained to implement a major module in this phase of the project. However, to minimize friction and getting in each other's way, we divided this project phase into several distinct and well-delimited modules, which were assigned locally or to the offshore team as appropriate.
- A two-week long on-site session of technical training, workshops, and coaching was organized for the offshore team. By the end of the session, the team was well-prepared and informed of what was expected of them, and a solid 'esprit-d'équipe' had been established.
- The main offshore QA engineer was brought over to work with the chief functional analysisist to study client requirements and testing strategies, and to get to know the European team better.

In phase 2, the same techniques that rang the alarm bell in the first phase : code reviews, deliverable-oriented testing, and reporting, and frequent milestones, now provided clear and objective evidence that the project was back on track.

Conclusion : Lessons learned

Nowdays, modern technology provides an abundance of ways to communicate efficiently at a distance : mail, chat, IM, IP telephone,.... Nevertheless, it can sometimes be difficult to get people to work and communicate efficiently if they have never physically met one another, especially if they come from different cultures. In today's distributed teams, this aspect of team management should not be overlooked. If the whole team cannot be united for budgetary purposes, as was the case here, key players should at least get to physically meet and work together.

It is also important to divide work in such a way as the work packages be as independent and as modular as

possible. Dependencies between the modules need to be well defined, and responsibilities clearly defined between the different teams. With good project management, distributed teams can work together well, but if certain work packages or modules need several people to cooperate closely, it will be easier if the people are on the same site.

About the Author

John Ferguson Smart is currently Project Director at Aacom, a French IT firm specialised in J2EE solutions. He holds a PhD in Computer Science from the University of Aix-Marseille, France. His specialities are J2EE architecture and development and IT project management, including offshore project management. He works on large-scale J2EE projects for government and business and writes articles about J2EE and project-management-related subjects at <http://www.jroller.org/page/wakaleo>.