Outsourcing and Globalization in Software Development

Jacques Crocker
UW CSE Alumni 2003
jc@cs.washington.edu

Slides available at http://seven.maqsoftware.com/cse403_outsourcing.ppt

---

Agenda

- Introduction
- The Outsourcing Phenomenon
- Leading Offshore Projects
- Managing Customers
- Offshore Development Notes
- Q & A

Introduction – Jacques Crocker

- Graduated with Bachelors in Computer Science less than 2 years ago from UW
- Took CSE 403 in Spring Quarter 2003
- Worked at a medium sized dotcom (Onvia.com) for 3 months after graduating
- Quit Onvia and joined a small outsourcing company with 10 employees in Seattle, and over 50 in Bombay
- Current position: Lead Software Engineer / Architect
- MAQ Software (www.maqsoftware.com)
- Currently working with 3 other UW CSE grads (David Owens, Ben Handy, Susan Shinoda)

Outsourcing

- What is outsourcing?
  - Hiring an external company specialized in software development to implement software projects
- Why outsource?
  - For companies with limited development needs, it's usually more cost effective to hire an experienced software team than to try to hire employees or contractors and manage them
- How are the contracts structured?
  - Projects are defined from the start as fixed cost or billed per time/materials

---

Outsourcing Alternatives

- Full-time employee dev teams
- Contract consultants (W2)
- Off the shelf packaged products

Outsourcing here to stay

- Key efficiencies can be gained in hiring a specialized team of developers who can ramp up quickly and get your project completed for a fixed cost
- Customers get exactly the application they want, own the code-base delivered, and have the flexibility to manage the direction of their product without the baggage of managing employees

Outsourcing Across the Globe

- #1 Reason – Cost
  - Developers in India cost (50-70%) less
- Other Reasons
  - Around the clock development schedule (time zone differences mean there can be 2 teams working on the same project, night & day)
  - Scalability – enables a small outsourcing firm like us to easily scale up and down according to the currently available workload

Major cost benefits?

- Does 50-70% cost savings for developers translate into 50-70% cost savings on the entire project?
  - Not quite
  - High communication overhead
  - Lower project visibility
- Offshore outsourcing significantly increases risk of project failure
  - Requirements not understood
  - Lower quality code
  - Final product potentially unusable

Offshore Development

- **Result:** applying risk reducing software engineering best practices are even more important when managing offshore development projects than in normal situations

Leading Offshore Projects

- Since the customers are in the US, there is always the requirement for a skilled local resource to manage the project end-to-end
- Get to do the fun stuff
  - Design the product
  - Architect the solution
  - Plan the implementation
  - Keep tabs on development progress
  - Finalize and deliver product to customer

What About Coding?

- While sometimes tempting when managing a software development project, don't stop coding!
- One of the key practices I've seen in successful offshore project management is active coding involvement of the local project lead
- One essential risk reducing practice is to review all code that has been checked into the project on a daily basis
- Writing Unit Tests on Business Components to automate build verification is a great practice to understand the code and provide valuable feedback

Offshoring Best Practices

- Some techniques to follow to significantly increase the chances that a project will be successful
  - Clickable Screens
  - Daily Code Reviews
  - Daily Build and Smoke Tests
  - Unit Testing

Clickable Screens

- Prototype the final product early on in the project to define what you are building
- Can use HTML, Image Maps, Flash, Rapid dev tools, depending on the type of project
- May go through multiple phases (post it notes, to photoshop, to html)
- Final result however should be something that near exactly resembles the final product in terms of look, feel, and workflow

Daily Code Reviews

- Throughout the life of the project, it's essential to be on top of all code changes, especially architectural ones that impact the application's overall API
- Checked in code should be reviewed for readability, logic, and functionality
- Bad coding practices need to be followed up on and addressed quickly

Daily Build and Smoke Tests

- After the first testable build milestone, every day the entire code-base should be integrated, built, and verified
- Don't let projects go more than 2 days with a broken build!
- Have this done offshore also so they have a chance to fix any major problems before they send it
- Either way, it's always required to rebuild the application every morning from the source tree and run it through the build verification unit tests to catch errors early on

Automated Unit Testing

- Automated unit tests are invaluable for testing the architecture and providing a resource to help others understand how to use the applications API
- Writing unit tests against other's code is also a great way to help understand it
- Keeping a growing library of unit tests will keep builds stable and clean
- Writing unit tests to demonstrate major application bugs will not only help the developers fix it, but also to ensure that the bug doesn't creep up again later

Managing Customers

- Interacting with customer / project stakeholders is a big part of managing offshore projects
- High project visibility = better feedback
- Main task here is communicating progress so they feel comfortable that the project will be successful

Customer Best Practices

- Some techniques to follow to manage customer and keep them happy with the progress
  - Clickable screens (again!)
  - Offer choices, don't ask questions
  - Weekly status meetings
  - Milestones

Clickable Screens for Customer Approval

- Have clickable screens that fully demonstrate the application functionality to show to the customer
- They'll be able to visualize the final product early on and more likely to give valuable feedback
- Increases their confidence that the project will deliver (much less likely to cancel the project!)
- While the screens will always be changing even after they sign off on them, having the screens approved before implementation starts will at least give some leverage in case you need to refuse a change that will significantly increase dev effort

Don't ask open questions

- When requiring customer feedback, it's almost always better to offer choices
- If you ask a question: "How should this report look?" you'll very rarely get a decent answer
- Usually you can piece together enough information to come up with something that makes some sense. Show it to them and they'll definitely tell you what is wrong with it

Milestones

- Break projects up into well defined milestones that list the functionality that will be delivered
- At each milestone, demo the app with potential application users and project stakeholders to get feedback on the updates to the build

Offshore Development Notes

- Communicating with your dev team
- Managing a test team
- Common mistakes / tendencies
- Future of the Industry

Communication with Offshore Dev Team

- Nightly conferences at the beginning of their day
  - Short conference call via phone to quickly discuss daily
  - Longer chat via Instant Messenger to go through critical bugs or architectural decisions

Communication with Offshore Dev Team (continued…)

- Daily status update emails at the end of their day
  - Report on progress completed for the day
  - Any pending issues to be discussed
  - Build information
- A robust bug tracking application is essential for tracking issue resolutions

Managing an Offshore Test team

- A very important part of every project is an organized and disciplined testing effort.
- Testing is one of the fitting capabilities to be offshored, even for local dev teams. Time zone differences make things very convenient

Open Discussion

- Questions?
- Thoughts?
- Opinions?