

Policing Offshore Software Development

Reduce Risk by Managing a “Good Cop and Bad Cop” Relationship

By Greg Spehar (spehar@full-knowledge.com)

Executive Summary

Distributed development and off-shoring initiatives have been promoted over the past several years as the mechanism for which software development teams can off-load work that is routine and does not require close contact with customers. With this offshore mechanism everyone knows that you can achieve a 30% to 50% cost reduction. So why isn't everyone doing it? The main problem involves risk. The risk is that the code will not be developed to the required standard, which will result in costly errors both at the testing level and at the field level. The Static Analysis solution overcomes the risk problem through a combination of tools that allow organizations to establish and enforce coding standards. Using Static Analysis tools customers can enjoy the cost benefits of outsourcing while at the same time mitigating the primary barrier to outsourcing – namely risk.

The following is an interesting data points for understanding the future playing field in software development and the use of offshoring:

“Forrester Research estimated the 3.3 million US services jobs will be relocated abroad between 2003 and 2018, accounting for US\$136 billion in wages. Over 400,000 of these will be IT related, with the greatest level of outsourcing expected in software development and customer service/call centres.”

A Fine Balance: The Impact of Offshore IT Services on Canada's IT Landscape by Robert Scott and David Ticoll, (This white paper is available upon request)

This trending in the IT arena also applies to the development of software and the overall impact to the organization that has to handle this new environment. This requires a new set of tools and processes that allow the code, under developmental change, to endure a rigorous quality assessment (policing) to verify the code has been developed to a high standard of quality reducing the risk of substandard work being performed. The policing of code is very time consuming if done manually and can be error prone if done repeatedly. So the following questions need to be answered:

- How can we police the code and ensure that the people developing the code are not alienated?
- How can the policing be done to ensure a consistent “good cop” assessment?
- What tools are needed?
- What processes need to be changed?
- How will the organization benefit financially?

Target Companies:

- IT Organizations
- Software Companies
- Embedded Systems

Target Audience:

- Executives
- Directors
- Software Managers

Our Goal:

Creating wealth by assisting in delivering your software projects to market fast and reliably.

Our Guarantee:

We guarantee that we will create wealth with a ROI that will exceed 1000%, or the services will be one third (1/3) the price.**

** Only applies to the wealth creation package.

Contact:

Full Knowledge LLC
2477 Santa Rita Rd #32
Pleasanton, CA, 94566

Office: 925-484-8490
Cell: 503-332-3663

Email:
sales@full-knowledge.com

Bad Cop

Creating a working relationship with any part of your company or any vendor that is critical to the maintenance or management of your software can make or break the viability of that relationship. ITMSA has done research that would indicate that a good working relationship with the offshoring group (or with offshore components of your company) could take some trial and error. And the efforts toward making the process work will in the end pay off.

“Experiences with offshoring have been mixed—most are neutral; just as many respondents had good experiences as had bad experiences. The bad experiences are most often related to communication and quality of service. Smaller companies report more negative experiences. Despite the negative experiences, buyers do not appear to be deterred. They will pick themselves up, dust themselves off, learn from their mistakes, and try again.”

Buyer Views on Outsourcing and Offshoring: Marketing Implications for ITO and BPO Providers

ITMSA Research Report - http://www.itsma.com/research/research_lr.htm

In short, the potential to always be the “bad cop” is apparent and real. There is a need to police the groups to the appropriate level such that the quality of the output is clearly as good or better than the products at the home site. This need is the clearest area of risk by offshoring. Moving software offshore might save money in the short term, but if the software product has more defects and becomes ever more complicated the risk is not worth the effort.

Enforcement of Software Development: CIO's must deal with Enforcement of Software Development and Best Practices while lowering cost. There are software tools that help monitor friendly and not so friendly (companies) nations. This is really an issue of standards and enforcement. That is why we use Pattern Based Design and Development Software Tools for global applications.

Michael Atkins

Chief Executive Officer, CFC

Should We Outsource Programming to Less-Than-Friendly Nations?

From a response to this online article at CIO.com –

<http://comment.cio.com/comments/11065.html>

So it is clear that to manage the relationship appropriately the need to be a “bad cop” is present to ensure the “enforcement” of standards and rules are applied to ensure the highest quality product creation and management. But the risk of becoming a “bad cop” all the time requires a consistent set of tools and processes, which can allow for the “good cop” to appear on the scene.

Good Cop

The use of processes and procedures has been used by many organizations to some level of success in offshoring companies such as those in India. These strict adherences to rules and procedures can produce surprisingly good software. The importance of ensuring the correct processes and procedures along with the right level of a pilot project can further assist in the adoption as well as reduce the risk of failure.

“...certifications are a good indication of the maturity of processes and systems within offshore software development companies.”

Best Types Of Projects For Offshoring
WebProNews.com

By Marty R. Milette

<http://www.webpronews.com/enterprise/enterpriseonline/wpn-13-20030813BestTypesofProjectsforOffshoring.html>

But over emphasis can also bring the ability to respond quickly to a grinding halt and place the team in chains and a prison of process they are unable to escape.

"The CMM demands lots of documentation and traceability procedures, very intensive quality assurance procedures and process support people, so there's a lot of resistance from managers," he says. "But setting up a facility from scratch and introducing the right practices from the very beginning is a good way to get a software development organization to a very high level."

Gary H. Anthes and Jaikumar Vijayan, Computerworld

<http://www.computerworld.com.au/index.php/id:233675276;relcomp:1>

Thus finding the tools necessary to allow for the flexible implementation of coding process that can supply a high quality and conformant product is critical. With this type of tool set the “good cop” can manage the offshoring development team, where the “good cop” assists in the enforcement of the coding standards and architecture in an automated manner. Where the home development and quality assurance team is able to perform the same exact automated tools to ensure that the offshore team is performing the work properly, thus allowing them to play the “bad cop” without destroying the relationships that are needed to complete the project.

The following sections will describe the tools that are needed and how they are deployed to ensure the “Good-Cop, Bad-Cop” effect is realized.

The Product Tools

There is a set of tools provided by Static Analysis Vendors that can provide an accurate set of feedback to reduce project risk, to ensure the advancement of the code base within quality standards and to ensure the correct coding policies are followed. This set of products require the following framework to perform their jobs properly:

- Identification of Issues – Static Analysis
- Understanding the Code Base – Architecture
- Protecting from Further Erosion – Desktop Static Analysis
- Measuring Risk and Code Progress – Metrics Management

Each of these products are intended to supply the proper controls to allow the offshoring team to manage their progress under the “good cop” enforcement strategy where these very same tools can be used at the home development site by managers, architects and quality assurance personnel to enforce the desired results and supply a mechanism to be the “bad cop” without risking the relationships required to complete the coding.

Static Analysis

Build time tool that runs at every build or at an appointed build to provide a snapshot of the system at a moment in time. The use of this build snapshot allows for the integration and execution of rules that will catch red handed all security and coding defects, metrics and architectural violations of the standards that were agreed upon prior to the development effort.

Architecture

Architecture tool that allows architects to see the actual relationships between files and allows the architects to perform what if scenarios and escape plans as they try to dig their ways out of architecturally unsound code and problematic code. Once they have bailed themselves out of the problems that were discovered, the architects can use the tool to define an architecture that is enforced by each developer and at every build.

Desktop Analysis

The Desktop tool allows a developer level enforcement to ensure the developer does not violate the defect controls for hard to find bugs and security defects. The tools can also track and warn the developer when they have passed metric thresholds that will put their code at risk down the road. And finally, the enforcement of Architecture must happen at the developer’s desktop. With the use of InSight the architectural intent can be translated to the developer such that they will not break the architecture laws that have been established.

Metrics Management

Metrics Management is the tool that provides detailed metric data build over build. This tool will give the developer and managers at the offshore site or distributed site the visibility and ability to gather information at the scene of the crime. Thus providing that information to the home site to back up their development decisions when the time crunches occur. The home site, having access to the same data, can also provide incentives that have clear metric objectives and goals allowing effective risk management.

The Process Changes

The use of the tools needs to be integrated into existing processes or additional processes need to be defined to ensure the tools are used properly. Those process changes enacted by development roles will provide the mechanism for success. There are many ways to use these products to dig out of the prison of inflexible processes. The following are just suggested implementations by product.

Static Analysis Reports

Managers – Build reports detail the progress of the development effort, which can be reviewed
Architects – Monitor the build reports to identify architecture violations at a system level
Quality Assurance – Monitor and review defects to make decisions on quality standards
TeamLeads/Developers – Review defects and schedule for updating code based on the reports

Architecture Tools

Managers – Architect Reports are generated to monitor the progression of the architecture
Architects – Discover, Understand, Manage and deploy architecture rules and decisions
Quality Assurance – Monitor and review architecture to verify quality standards
TeamLeads/Developers – Monitor using InSight Developer architectural intent

Desktop Analysis

Managers – Monitors the development team to ensure the tools are performing
Architects – Performs periodic checks on the code using Desktop Analysis to ensure tools is performing
Quality Assurance – Monitors the usage of the Desktop tool by monitoring the Static Analysis Reports
TeamLeads/Developers – Enforcement tools used when making any changes to the code

Metrics Management

Managers – Monitor churn rates and make development decisions based on trending and risk data
Architects – Monitor complexity and risk points for indications of refactoring
Quality Assurance – Monitor and review churn rates, complexity and risk hot spots
TeamLeads/Developers – Monitor the metrics components and trending to ensure compliance

Where these tools are used by the offshore development site by the management at those sites so that they will have the ability to guide the quality efforts with a “good cop” mentality. The home site core product team and core architects can play the “bad cop” by enforcing and ensuring the processes and procedures coupled with the design decisions and standards are deployed and used appropriately.

Now that the products and processes are better understood, the financial benefit is shown to potentially exceed several million dollars for every million lines of code (MLOC). At 10% churn rates the value can exceed \$5Million alone, at churn rates exceeding 50% the savings can reach \$15Million plus. The following section will outline this financial value that can be achieved by playing “Good cop – Bad cop” effectively.

The Financial Benefit

To ensure a complete and concise evaluation of the benefit several assumptions will be made that may differ from the target environment.

Assumption 1: Development Costs

Code Reviews - \$100 per hour per person – Average \$500 per 1000 Lines of Code (LOC)

- Test Defects - \$500 per defect if found prior to testing
- Field Defects - \$15,000 per defect if found prior to deployment into the field

Assumption 2: Development size and change

Project Size: 1,000,000 LOC

- Offsite Churn Rate: 20% of the code
- Effective Defect rate of code: 6 per 1000 LOC (KLOC)
- Testing Defects caught by Static Analysis : 10% - 50% (Currently rules of thumb)
- Field Defects caught by Static Analysis : 1% - 5% (Currently rules of thumb)

Assumption 3: Use of metrics/architecture impacts included

The following would be the calculations to derive the benefit of these tools:

Code Review

First pass: 1MLOC / 1000 = 1000 hours of work * \$500KLOC/hour = \$500K

- 1) Churn Rates: 20% per year = 200KLOC/1000 = 200 hours * \$500KLOC/hour = \$200K
 - a. Effective IRR of 10% would provide net present value of \$2M
- 2) Total value is \$2M with churn plus one time run or \$500K = \$2.5 M per MLOC

Finding Defects prior to Testing

Initial pass: 1,000,000 LOC / 1000 = 1000 * 6 defects = 6000

- a. Per million lines of code there exists 6000 defects today
- b. Static Analysis may find 10% of 6000 = 600 * \$500 per defect = \$300K
- c. Static Analysis may find 50% of 6000 = 3000 * \$500 per defect = \$1.5M
- 1) Churn Rate of 20% assumed is 200,000 LOC / 1000 = 200 * 6 defects = 1200
 - a. Per million lines of code with churn of 20%, 1200 defects are added per year
 - b. Static Analysis may find 10% of 1200 = 120 * \$500 per defect = \$60K
 - i. Effective IRR of 10% would provide net present value of \$600K
 - c. Static Analysis may find 50% of 1200 = 600 * \$500 per defect = \$300K
 - i. Effective IRR of 10% would provide net present value of \$3M
- 2) At 10% the total value is \$600K with churn plus one time \$60K = \$660K per MLOC
- 3) At 50% the total value is \$3M with churn plus one time \$300K = \$3.3M per MLOC

Finding Defects prior to Field

Initial pass: 1,000,000 LOC / 1000 = 1000 * 6 defects = 6000

- a. Per million lines of code there exists 6000 defects today
- b. Static Analysis may find 1% of 6000 = 60 * \$1500 per defect = \$900K
- c. Static Analysis may find 5% of 6000 = 300 * \$1500 per defect = \$4.5M
- 2) Churn Rate of 20% assumed is 200,000 LOC / 1000 = 200 * 6 defects = 1200
 - a. Per million lines of code with churn of 20%, 1200 defects are added per year
 - b. Static Analysis may find 1% of 1200 = 12 * \$1500 per defect = \$180K
 - i. Effective IRR of 10% would provide net present value of \$1.8M
 - c. Static Analysis may find 5% of 1200 = 60 * \$1500 per defect = \$900K
 - i. Effective IRR of 10% would provide net present value of \$9M
- 3) At 1% the total value is \$1.8M with churn plus one time \$900K = \$2.7M per MLOC
- 4) At 5% the total value is \$9M with churn plus one time \$4.5M = \$13.5M per MLOC