

Monitor, Measure and Control Outsourced Development

Bringing transparency to outsourcing

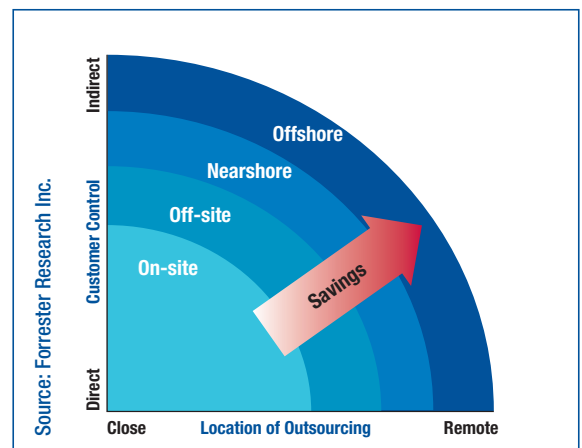
SOLUTION



Outsourcing: the Silver Bullet of Application Development?

Companies have long understood the benefits of outsourcing: reduced costs and improved service levels through greater IT performance and flexibility. "Cost reduction remains the primary motivation behind current outsourcing contracts, but an increasing number of companies - 21% now versus 11% in 2004 - are outsourcing primarily to improve quality" ¹. Unfortunately companies often tend to overlook the associated risks and according to a recent study by Deloitte Consulting, 70% of senior executives reported significant negative experiences with outsourcing projects.

Still the most progressive IT organizations - large and smaller ones alike - continue to take a cautious but forward position toward outsourcing.



Picture 1: Outsourcing models

¹: "Outsourcing saves less than claimed", Reuters, April 13, 2006



Visibility brings confidence

Automated Code review for greater visibility

The CAST Application Intelligence Platform, relying on advanced source code analysis technology, is designed to help AD organizations manage and measure supplier deliverables. The CAST Application Intelligence Platform automatically analyzes the health of applications using multi-parameter metrics and indicators – broken down by quality & quantity categories – so that you know what your outsourcer is delivering to you and your business. CAST provides automated monitoring and control of the maintainability, technical quality and costs of the applications being delivered so that you get the visibility to audit and challenge those deliverables.

Innovative companies like the Depository Trust & Clearing Corporation, the US Food and Drug Administration, SFR-Cegetel, and Deutsche Telekom are using a small portion of their savings from outsourcing to ensure that poor quality doesn't impact their long term cost of ownership and future business success.

Gain control over the deliverables from your outsourcers

With CAST, you can automatically monitor compliance with development best practices and architectural guidelines. In fact IT managers can easily measure applications against a set of quality rules pertaining to architecture, documentation, performance, naming conventions, best practices and more while looking at the trends across teams or the same teams over time.

Source Code Quality – the hidden risk of outsourcing

From legacy mainframe applications to new distributed Internet-based programs, companies are relying on External Service Providers (ESPs) to build a network of offsite, nearshore and offshore competencies to maintain and deliver reliable systems at the lowest possible cost.

Over-engineered programs become unnecessarily complex, making the resulting software harder to maintain but also making programs difficult to understand and hard to learn for new resources who may be assigned to maintain and enhance them. The client ends up having a greater than expected dependence on the software writer.

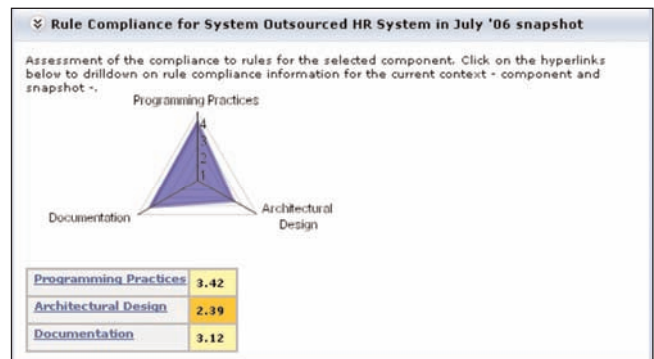
With systems becoming increasingly complex, the situation can easily lead to decreased quality, delayed deliverables, insufficient deliveries, heightened security risk, vendor lock-in and increased maintenance costs when key development and architectural practices are not fully followed by the outsourcing partner or when the knowledge transfer is not efficient enough between both parties.

Companies often rely on Service Level Agreements (SLAs) that focus almost exclusively on testing performance and function without any requirements for the quality of the code itself. So what is the result of relying upon inadequate SLAs to manage (even well-intentioned) outsourcers?

This all translates into one thing: greater risk.

Sophisticated organizations are automating the analysis of supplier deliverables to ensure they can answer questions like:

- Are there security holes?
- Is the vendor adding insecure or even malicious code to your applications?
- How well are architectural best practices followed?
- Are they abiding by coding standards?
- Is the code well documented?
- Have development frameworks been followed and utilized correctly?
- How easily can the application be maintained and changed in the future?
- Can development be brought back in-house or switched to another provider quickly and easily in the future?
- And many more...



Application Health Factor	Transferability	Changeability	Robustness	Performance	Maintainability (SEI) Security
Outsourced HR System - Nov '05	3.32	3.37	3.37	3.77	3.18
Outsourced HR System - Jan '06	3.38	3.43	3.46	3.85	3.18
Outsourced HR System - Feb '06	3.24	3.35	3.39	3.74	3.09
Outsourced HR System - May '06	3.21	3.33	3.37	3.7	3.04
Outsourced HR System - July '06	3.21	3.33	3.37	3.7	3.04

Rule Compliance	Programming Practices	Architectural Design	Documentation
Outsourced HR System - Nov '05	3.44	2.26	3.37
Outsourced HR System - Jan '06	3.51	2.41	3.37
Outsourced HR System - Feb '06	3.44	2.4	3.16
Outsourced HR System - May '06	3.42	2.39	3.12
Outsourced HR System - July '06	3.42	2.39	3.12

Picture 2: Development Team Compliance with Best Practices

The respect and correct use of existing development frameworks, naming conventions and programming best practices all result in applications that are much easier to maintain and more easily understood by new team members.

This means that the source code must be both readable and transferable. The CAST Application Intelligence Platform measures this by evaluating in priority the:

- Level of compliance to Architectural and Design standards
- Level of compliance to Naming Convention & Coding Style rules
- Availability of Code Documentation
- Component's Size and Complexity

And you always have access to CAST Technical Knowledge Base if you eventually plan to in-source the application or safely transfer it to another outsourcer.

Streamline knowledge transfers

While code documentation is clearly identified by managers as being key to the knowledge transfer process, code size and complexity is usually largely underestimated as a barrier to the knowledge transfer and acquisition process. The CAST Application Intelligence Platform provides knowledge transfer automatically to baseline the application portfolio providing qualitative and quantitative metrics, helping identify those applications at risk which need to be completely documented. Then, with CAST, you can generate detailed technical documentation of the modules at-risk or applications where the risk has been identified because of:

- technical concerns (size, quality or complexity of applications)
- critical functional concerns
- or even key application subject matter experts are expected to leave or are no longer part of the team

As a result, the technical knowledge transfer is accelerated by up to three times on the newly acquired applications, and the risk of future delivery failures is reduced simply by enhancing the quality of the technical knowledge transfer. Since CAST collects metrics based on a "DNA"-level analysis AD Management can drill down and provide automatically-generated fresh technical documentation as needed to the development team.

Become a champion at Outsourcing

CIO.com reports in a February 2006 article that large-scale outsourcing deals promise big savings but still fail half the time. The golden rule of outsourcing according to Gartner: the decision to outsource should be rationalized, support your overall strategy - by aligning your application investments with your business objectives - and be built on a plan for long-term flexibility and reasonable expectations. Only companies with a solid outsourcing business case and strong partnerships with their outsourcers will manage to leverage the anticipated cost savings. So who are they?

They're the ones who managed to stay in control. They kept their intellectual capital in-house while outsourcing some of their application development and maintenance to a third party vendor, mitigating future potential risks. They streamlined the knowledge transfer process with automatically-generated fresh technical documentation. They improved their bargaining position with their outsourcers by truly assessing their performance with key metrics and benchmarking techniques. They made sure that software development best practices were respected and that quality work was indeed delivered. And in some cases, they were able to renegotiate their contract or even reconsider their decision to outsource and secure their exit.

At CAST, we've helped many of them and are continuously working to provide the best application development solutions your business really needs.

CAST in ACTION

Depository Trust & Clearing Corporation (DTCC) is jointly developing with a software development partner, business-critical applications for underwriting and corporate actions processing which support \$60-\$70 billion worth of transactions daily. The applications will be comprised of approximately 10 million lines of code across C++, mainframe and database languages. While DTCC employs multiple applications and methods to ensure the security of its software, with a job so important to DTCC's business, it is imperative that DTCC understands what its strategic development partner is delivering.

DTCC's three key objectives in controlling development are to:

- Develop applications that meet the critical needs of the business
- Ensure that its strategic development partner delivers high quality code
- Guarantee that the new applications will be maintainable in-house after development is completed

The CAST Application Intelligence Platform has given DTCC a solution that can understand, validate and report on all of the technologies used within the application including 3GL, Mainframe and database languages. CAST enables DTCC to automatically review every line of code delivered for technical quality, architectural standard compliance and security flaws.

With CAST, DTCC can ensure that the application is of good quality and that the transition from development to in-house maintenance will be smooth and problem free.