

CAST AD Performance Management Software helps software vendors efficiently deliver high quality software, facilitate and discipline customization work by clients

Combining efficient delivery of high quality software and customer satisfaction is a tough challenge

Customers have two major requirements for software vendors about application software:

- Very high technical quality: stable, bug-free, well performing, scalable, secure, and well architected
- Closest possible match with their functional and technical needs.

Faced with the rapid pace of change in development techniques and with increasingly complex development organizations, modern product groups must demonstrate excellence in their development practices. They must also provide good support for software customization efforts by the “customization supply chain” that includes Clients, their Systems Integrators and/or Vendors’ own Professional Services. Good support entails two things: First,

With CAST,

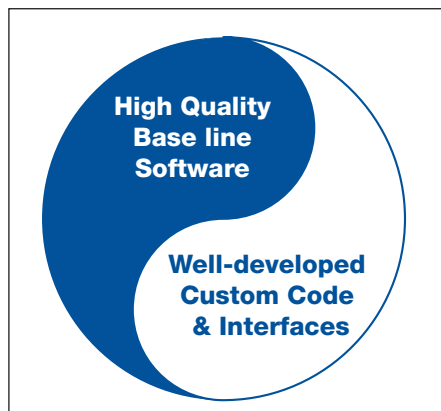
- **Product Groups:**
 - Deliver software with less bugs, better performance, security and maintainability
 - Improve productivity, predictability, and time to market
 - Better manage distributed and offshore development organizations
 - Adapt faster to new architectures (SOA...)
 - Better Document applications, APIs and version changes for customers
- **Clients, their System Integrators, or Software Vendors’ Professional Services:**
 - Better understand interactions between software base line and custom applications
 - Tightly control compliance with Vendor’s best practices for custom code
 - Precisely identify software version or patch changes and explore their impacts
 - Better monitor internal or outsourcer dev. team performance and quality of delivery

facilitating custom code development and creation of interfaces with other enterprise applications. Second, helping enforce development discipline to avoid deterioration of the overall technical quality of the customized system.

The need to better manage Product Group performance

Products Groups are required to rapidly turn out masses of new functionality, while at the same time, they have to restructure software to create customization APIs, SOA, or merge software resulting from acquisitions. They must achieve this without affecting stability or performance of the code base; and they must do so in increasingly complex development organizations, with multiple development sites, often offshore or outsourced. Confronting this challenge requires product groups

driven by performance and operational excellence. Since you can’t manage what you don’t measure, excellence starts with automated ways to objectively measure Application Development Performance of groups or individuals. This requires automated measurement of their output, of the quantity and quality of their software delivery and answers to questions like: How much did a team produce - functionally and technically - between two versions? Have Key Application Health Factors (KAHF) improved or deteriorated? Is development compliant with Development and Architecture rules? How does team or individuals’ performance evolve? How does it compare to other teams? Of course, measurement is not enough, development teams need action-oriented information to actually help them efficiently implement corrective action and improve their performance.



The need to facilitate and discipline software customization

Despite vendors efforts to provide functionally-rich and easy-to-adapt software, the reality is that many customers end up developing large amounts of custom code, either to extend software functionality or to interface it with other applications. This exposes both clients and vendors to the risks of impoverished quality of the resulting customized applications - with stability, performance and maintainability issues - and future difficulties in upgrading software. The situation may be aggravated by the fact that much of custom development is often done by System Integrators, who may not be in a position to prevent it. Often, this puts Vendors' Professional Services under significant stress and results in customer dissatisfaction.

How CAST helps improve AD Performance

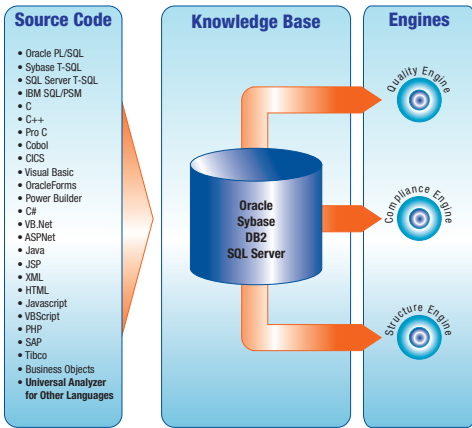
How CAST works

The CAST Application Intelligence Platform acts like an automated code inspection system – looking “inside” to identify code and architecture quality issues, to monitor development performance and to document applications. CAST’s Application Intelligence Platform can read, analyze and semantically understand all major kinds of source code, across all layers of an application (GUI, logic and data). By analyzing all tiers of complex software, critical application health metrics like robustness, maintainability, transferability, performance or security can be measured and compliance to programming best practices can be assessed. Furthermore detailed technical information about application structure may also be provided to teams.

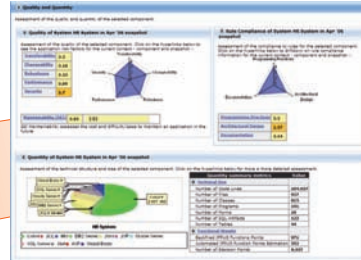
Actionable information for managers and their teams

With Application Intelligence generated by CAST and displayed in a portal (the AD Governance Dashboard), IT managers can:

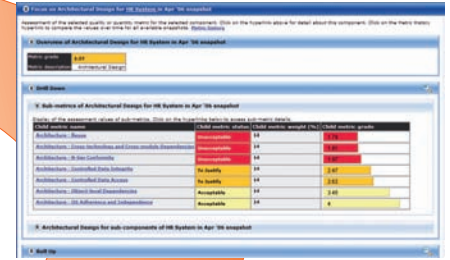
- Identify application issues before they become production problems and drill down directly to the underlying code at risk to help speed remediation.
- Preserve and enhance the quality of their applications assets for reduced maintenance cost and increased stability and performance
- Better manage the activities of local, remote and outsourced development teams, by measuring and benchmarking the quantity and quality of work delivered, and by verifying the justification for workloads and costs.
- Benefit from the enhanced productivity, predictability and risk reduction resulting from the empowerment of internal and external teams with technical documentation and application structure exploration capabilities.



Measure Quantity and Quality delivered by dev teams & Compliance with Dev. & Architecture Rules



Drill down to identify key factors causing quality shortfalls



Some of CAST automatically-generated Application Intelligence

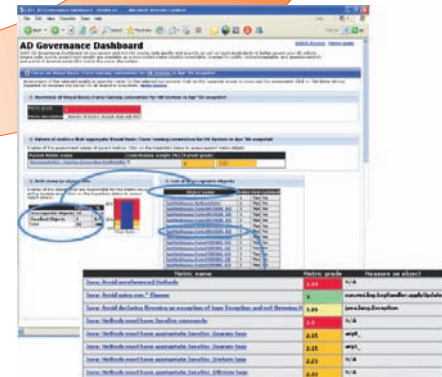
Detect version changes or changes between base line and customized apps
Verify what changes should have cost

Content Evolution Summary			
Legacy HR application Feb '06 Variation (% or details) vs 2005			
Functional Weight	158.08	222.92	141%
Number of Functions Points	1,651	4,173	252%
Technical Size			
Number of Code Lines	25,484	38,977	152%
Number of Artifacts	223	66	29%
Number of Processed Objects	66	(32/4/0/30)	
Added (L/M/N/VH Complexity)	66	(32/4/0/30)	
Updated (L/M/N/VH Complexity)	0	(0/0/0/0)	
Deleted (L/M/N/VH Complexity)	0	(0/0/0/0)	
Effort			
Estimated	21		
Estimated	142.94	(32/4/0/30)	
Addition (L/M/N/VH Complexity)	142.94	(32/4/0/30)	
Updated (L/M/N/VH Complexity)	0	(0/0/0/0)	
Deletion (L/M/N/VH Complexity)	0	(0/0/0/0)	
Effort/Number of Processed Objects	0.32		
Cost			
Estimated	2.17		
Estimated	1,602	(32/4/0/30)	
Addition (L/M/N/VH Complexity)	100,030	(32/4/0/30)	
Updated (L/M/N/VH Complexity)	0	(0/0/0/0)	
Deletion (L/M/N/VH Complexity)	0	(0/0/0/0)	
Cost/Number of Processed Objects	24.27		

Empower dev Teams to make changes with live technical documentation on custom code, base line code and their interactions



Provide dev teams with lists of defaults to correct



Impact-free implementation

This bounty of objective and actionable information helps managers and their teams improve AD practices without changing anything to the processes or the Application Life Cycle Management (ALM) Tools already in place. Application Intelligence simply is must-have additional information that contributes to AD performance improvements at all levels of the IT organization. Just like Business Intelligence helps managers make better business decisions, Application Intelligence from CAST helps AD managers better manage their AD team performance and improve it. Managers assess, monitor and benchmark applications and team activities and dev teams discover, explore and share detailed information about application quality, structure and the impact of changes.

Practically speaking, the CAST Knowledge Base is implemented in a Center of Application Intelligence where power users of CAST manage the analysis of applications, the development of ad-hoc reports and metrics, and the availability of specific application documentation. Managers and dev teams only need to look at the information they need in the CAST Portal. When team members need context-specific information like detailed documentation about a process or transaction, or an impact analysis to assess consequences of a change, they ask power users in the center to run it for them.

Improving Product Groups Performance

CAST helps Product Group managers in several key areas:

■ Preventing more production bugs, performance, security or maintainability issues

It is very difficult to catch all bugs before applications go live in production and are delivered to customers, especially when these bugs originate in architecture-related issues. Testing can hardly catch everything unless enormous resources are devoted to it. Furthermore, it is well documented that default corrections caught late in the development cycle - just before production - cost 10 to 20 times more than those corrected during development. Last, lack of attention to key application health factors eventually results in their drift as applications evolve, resulting in higher

production risk and increased maintenance costs. CAST helps pro-actively manage application robustness, performance, security, transferability or maintainability and detect and correct defaults early in the dev cycle, when they cost less.

CAST does so: By detecting many damaging coding and architecture practices early, right after they have been developed; by listing defaulting application objects; by highlighting potentially improper code lines in each object; by providing teams with technical documentation and architects with in-depth application discovery tools to facilitate corrections. CAST also helps improve testing practices by measuring the testability of applications and components, by calculating the number of test cases necessary and by helping eliminate unnecessary test scenarios through support to impact analysis.

CAST implementation typically results in a 5 to 15% decrease the occurrence of critical production bugs, a 10%+ reduction in corrective maintenance costs, and lower stress on client support teams.

■ Improving team productivity, predictability, time to market

Many factors contribute to sub-optimal productivity and predictability: Lack of objective metrics about application maintainability and about efforts requirements for change impact mitigation usually result in improper resource allocation and in costly maintainability drifts over time. Lack of technical documentation results in poor application knowledge transfers and sharing, in exposure to team turnover, and in unnecessary rework. *CAST helps: Measure the quantity and quality of delivery at team or individual level; enables the monitoring of application maintainability as well as drift avoidance actions; it enables time-saving proactive management of technical quality; reduces application discovery time upon transfers or employee turnover with technical documentation; last, it facilitates change impact mitigation in terms of impact identification, workload estimates, implementation and testing. Altogether, the combined gains from all these improvements typically result in a 15%+ productivity gain in product groups' delivery.*

■ Monitoring and facilitating distributed and offshore teams performance

More and more, development organizations tend to be distributed across multiple locations, including offshore sites. Furthermore, some of the development may be outsourced. AD organizations are thus increasingly confronted to: Difficult application knowledge sharing and transfer between sites, enormous application knowledge losses offshore because of very high offshore team turnover, difficult and costly Performance Management of remote teams and high risks of poor quality of unmeasured delivery. *CAST helps reduce AD Organization exposure and augment their effectiveness with Web-based automated technical documentation of software - easy to use and share - and with automated measurement of quality and quantity of delivery from remote teams*

With CAST, unproductive knowledge transfer times are typically reduced 30 to 50%, notwithstanding the other gains already described above.

■ Facilitating software restructuring (API creation, SOA...)

Software Vendors can hardly afford to discard applications and restart projects from scratch. Application restructuring is thus a way of life, be it simple refactoring, creation of APIs to facilitate non-intrusive customization of base line software by clients, or the creation and maintenance of a Service Oriented Architecture. Restructuring projects are often risky and costly because of difficult identification of critical application objects and lack of understanding of structural change impacts. Further development of SOA requires the ability to enforce appropriate coding & architecture discipline from dev teams as well as the availability of proper technical documentation of the services already created. Otherwise much effort will be sunk into unnecessary rework and duplication.

CAST facilitates software restructuring with: Automated identification of critical application objects; documentation of interactions with other objects; automated detection of non-compliance with coding and architecture rules; and web-based automated technical documentation of software, easy to use and share.

Potential productivity gains in restructuring projects are well above 20%.

■ **Documenting applications or APIs and their changes for customers**

Vendors must enable base line software customization, extensions and interfacing. Whether they do so by delivering application source code to customers or by making structured APIs available to them, it is very clear that the software or APIs will evolve significantly version after version. Vendors need cost-effective ways to document each version of applications and APIs for customers and partners as well as the changes between versions.

CAST provides web-based automated technical documentation of software - easy to broadcast to customer in CAST portal (with appropriate licensing) - and automated identification of changes between versions.

Using CAST will result in 50% gain in complete technical documentation production, always up-to-date documentation, and very significant improvement in customer satisfaction.

Facilitating and disciplining custom development work by Customers, System Integrators and Professional Services.

Most customers of Enterprise Software need to customize base-line software beyond the built-in parameterization capabilities. They thus develop vast amounts of custom code either to extend software functions, or to interface it with other key applications. This work can be done by customers' own dev teams, more often by their Systems Integrators, and usually with some support from the software vendors' Professional Services. Software vendors must facilitate the work from all actors in this "customization development supply chain" and make sure it is done in a disciplined manner that preserves customized software technical quality and upgrades adoption. CAST helps in several key areas:

■ **Better understand interactions between software base line and custom applications**

Vendors' Product Groups may use CAST to automate delivery of technical documentation about software base line, APIs and their changes to customers. This greatly facilitates clients' effectiveness in customizing software. However, when customers need to develop additional custom code on top of already customized applications, or when they want to develop

complex interfaces between Vendor's Software and other applications, they need to understand interactions between base line software and other applications or commercial software. CAST provides clients' dev teams with automated web-based documentation and architects with in-depth application discovery tools for all applications they need to customize and interface. ? Using CAST will result in 50% gains in customization and interfacing development efforts.

■ **Tightly control custom code compliance with Vendor's best practices**

The flip side of facilitating customization is the risk of letting customers or their SIs develop vast amounts of custom code that may first lead to a degradation of the overall technical quality of the customized system, with stability, performance, security or maintainability issues for which customers may blame the vendor. Eventually software upgrades may have huge impacts on this custom code, leading to delayed adoption of the upgrades.

CAST's automated development and architecture rules compliance checks help verify custom code compliance with Vendors' recommended best practices. CAST also lists defaulting custom code objects and highlights potentially improper code lines in each object. Vendors can easily implement many vendor-specific development rules into CAST, or ask CAST to do it for them. Customers or Vendors' Professional Services may also use CAST to check the compliance of System Integrators' custom code delivery with the same rules.

CAST's use by any actor of the custom development supply chain results in significant gains in custom code maintainability, reduced quality risk, faster adoption of upgrades and increased customer satisfaction.

■ **Precisely identify software version changes and their impact**

When the amount of custom code on top of Vendor's Software is significant, Software upgrades or even patches may have huge impact on this custom code. This risk and the costs associated with upgrade impact mitigation efforts often result in delays in client adoption of upgrades or patches, leading to increased customer support costs and potentially deferred business.

CAST enables precise identification of software or API version changes as well as the identification of changes between base line software and customized software. IT then enables the analysis of change impacts on custom code or interfaced applications with documentation of interactions between custom and base line code.

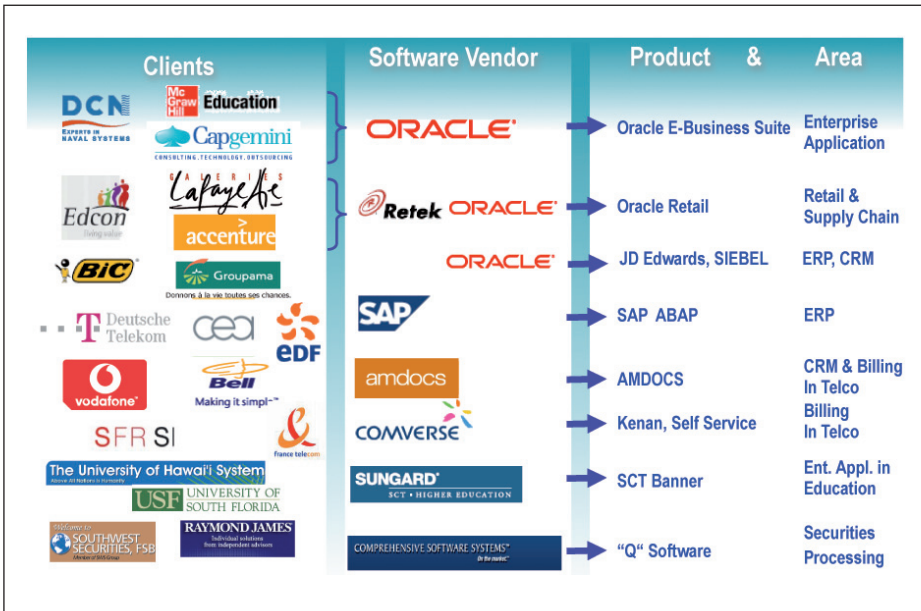
CAST's use by any actor of the custom development supply chain results in significant gains in upgrade costs, reduced risk, faster adoption of upgrades and increased customer satisfaction.

■ **Better monitor internal or outsourcer dev. team performance and quality of delivery**

The custom development work and its maintenance is often done by remote teams or System Integrators and the costs associated are significant. It is thus important for customers to be able to objectively measure the performance of these dev teams, to benchmark them and to challenge their costs. It is often in the best interest of the SI themselves to be able to cost justify their work, as its difficulty may be underestimated by customers. Furthermore, safeguarding key application health factors will avoid future drifts in their maintenance and support costs.

CAST enables precise identification of custom code changes upon delivery and automated estimates of how much these changes should have cost. Automated documentation and exploration tools also enable deeper understanding of the changes that took place. Automated technical quality and compliance checks enable precise measurement of the quality of these changes and helps avoid drifts. CAST's use by customers in AD performance measurement helps significantly reduce custom code maintenance and RFC costs.

Examples of CAST use on specific Software



CAST – The technology

CAST Application Intelligence Platform relies on unrivaled source code analysis technology developed over more than 15 years. CAST combines unique depth of analysis - allowing a fine-grained representation of application object structure and relations – to unique coverage of all major development languages, all combined into a unified description of entire applications and systems across multiple tiers and technologies.

CAST supports most major programming languages with dedicated analyzers:

- Client tier: HTML, Java, J2EE, .NET, C/Pro C, C++, Oracle Forms, ABAP, COBOL/JCL, Power Builder, Visual Basic, Delphi...
- Database platforms: SQL from Oracle, Microsoft SQL Server, Sybase, IBM UDB DB2
- Middleware and others: TIBCO, CICS, IMS, Business Objects ...

CAST also features Universal Analyzer, a programmable analyzer that allows rapid coverage of other languages and environments: It has been used to rapidly deliver customer value on software environments like: WebMethods, Uniface, JDEdwards, Siebel, PeopleSoft and many more.

CAST - The Company

CAST is the world leader and pioneer in Application Development Performance Management software, providing the metrics and information IT executives must have to measure, control and improve the quality of business applications and the performance of development teams around the globe. Founded in 1990, CAST has helped more than 600 organizations worldwide speed delivery to the business, mitigate risks in production, improve the customer experience, and reduce the total cost of application ownership.

CAST software is being used to industrialize the development of complex business applications and transform the development world – currently perceived as complex, obscure and sometimes artistic – into a world that is easier to understand, transparent and driven by performance and operational excellence. CAST is recognized as a market leader in its field by leading analysts like Gartner or Forrester.