Codagen™
The evolution of development

Client Success Story

CGI

CGI

When CGI needed to dramatically increase productivity during a multi-million dollar development project, they decided to take an MDA™ approach. CGI selected Codagen.

> "I never saw a tool that could really carry out the code generation I was looking for in the past. When we understood Codagen and realized it was completely different from the others in the market, it was like 'Wow!' We've been waiting for this kind of tool for years."
>
> – David Bertrand, Technology Architect, CGI

Founded in 1976, CGI is the fourth largest independent information technology services firm in North America, based on its headcount of 13,700 professionals. CGI provides end-to-end IT services and business solutions to more than 3,000 clients worldwide from more than 60 offices. The annualized revenue run-rate totals CA$2.1 billion (US$1.3 billion), and its order backlog currently totals CA$10.7 billion (US$6.7 billion).

The Mandate

The systems of one of CGI's largest customers would have to be re-created from top to bottom using new technology. The effort estimated for completion of this mandate is over 50,000 person-days, with delivery occurring within three years.

To address this sizable challenge, CGI has built on the expertise and synergy between themselves and the customer. The CGI team has implemented best practice-based methodologies and has adopted tools that yield increased levels of productivity in terms of overall project realization, development, analysis, testing, etc. Because there is a three-year window for development, both staffing and the development effort itself are mission-critical issues; however, CGI recognizes the strong probability that the technologies chosen will evolve, creating training issues as well as integration and interoperability restrictions. With strong insight into these challenges, CGI decided to use the Object Management Group's Model Driven Architecture (MDA) approach. MDA works by effectively separating the business logic of an application from the infrastructure in which it is deployed.

The Paradox

Approximately 60 people are working on the project. A team of six is in charge of the core development process, with the rest involved in the architecture, the requirements management, and the business analysis. More developers will be added soon, and most development will remain in house. Finding the required skills in the market has become difficult, and because Java programming is typically done manually, CGI has come face-to-face with an inherent paradox—How can they achieve the high levels of productivity required to meet the deadline but still keep the development process, architecture, design, ownership, and testing in house? They scrutinized the development process and decided that code automation was the solution. To increase productivity and accelerate time to market, CGI suggested Codagen, an MDA standards leader. That strategic decision would prove to save over 30% of the customer's development costs.

ROI CALCULATION	
Total Development Cost without Codagen	$8,073,450
Total Development Cost **with Codagen**	$5,474,700
Savings	**$2,598,750**
Cost Reduction	**32.2%**

The Race to Productivity

Four automation tools were available on the market. Using a proof of concept to conduct benchmarking tests on the J2EE platform, the company rated each tool on key criteria:

- Installation/ configuration
- Learning curve
- Ease of use
- Productivity gains
- Performance
- Flexibility
- Compatibility with J2EE
- Java compatibility
- Quality of code generated
- Cost
- References
- Potential ROI
- Client support
- Company momentum

The clear choice was Codagen's tool, Codagen Architect. After seeing CGI's organic architecture, the framework, and patterns, Codagen's specialists provided CGI with several pre-built, custom templates. "We implemented the tool with their help, and in a week, we were up and running," said David. "In terms of the learning curve, the implementation went so fast—we had one day of training and the transfer of the templates they built for us—the next week we were generating code."

Development Environment

CGI's development environment is deployed on a Windows 2000 platform and integrates a set of tools that support the development process based on Rational Unified Process (RUP) methodology. Rational tools are used for requirements management, configuration/ change management and modeling. Their IDE is IBM's WebSphere Studio Application Developer (WSAD, the new generation of Visual Age for Java/ IBM), and they use WebSphere Application Server as a J2EE application server.

For code generation, CGI has integrated Codagen Architect. After the business modeling is done at an analysis level, the design is realized, the link is established between the Rational Rose model and Codagen templates, which encapsulate all of CGI's architecture frameworks and design patterns, at which point JAVA code is generated. This is fully integrated and is now in production.

Faster Pace, More Control

CGI has found that one of the biggest impacts of working with Codagen is that they can encapsulate their J2EE framework within their Codagen Architect templates and generate code, so developers need only know how to code Java, not the J2EE patterns and frameworks. This puts CGI way ahead in the development game. If they had to develop everything by hand, everyone would have to know the J2EE framework.

Furthermore, CGI wanted a kind of "white box™" solution that would provide total control over the architecture, not a solution based on proprietary frameworks or business rules (Codagen Architect uses templates which provide total control of the generated code). CGI writes their own frameworks and design patterns and implements them in the templates; through the models, the tool generates all the code. "The only mistake I can make is if my template isn't that good…" stated David. "We have a layered architecture, and each layer has frameworks and patterns. Through (Codagen) we are able to generate 50–60% of the code—I have complete control."

"I had planned a month and a half to build the templates, but in two weeks, we were up and running."

– David Bertrand

"We've had great support [from Codagen]. The technical staff and technology are amazing. Everyone's working in the same way to ensure customer success. If there's a problem, they resolve it very quickly. It's a bit clichéd, but it's a win-win relationship between our companies," says David.

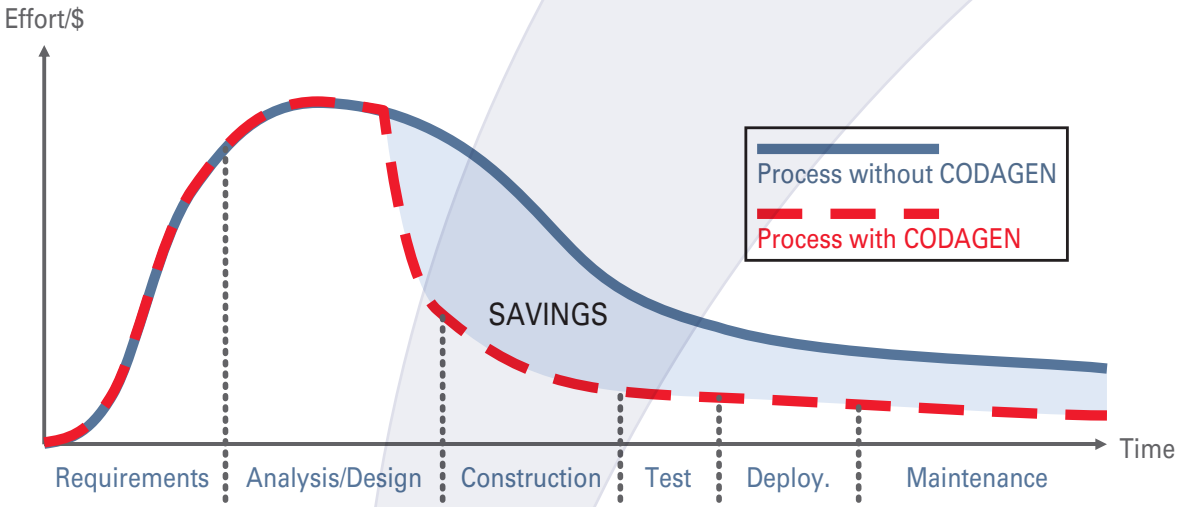
The Bottom Line: Results That Add Up

Given CGI's concerns about staffing, automating production of more than half the code has an enormous impact. "Now I don't need to seek J2EE experts, just Java developers. That saves a lot of money, especially since J2EE developers and programmer/analysts with practical experience are very rare on the market." With Codagen Architect, the J2EE patterns and frameworks become transparent to Java developers, and everything gets translated through the code generator. This makes customer's deadlines much more realistic—and results in serious savings.

CGI estimates their average cost for design, development, and testing at $77 per hour. Fewer software testers and developers will be required over several quarters in 2002 and 2003. Reducing the effort by one person for one week saves $3,080. In the second quarter of 2002, three fewer developers were needed, saving $110,880. In the first quarter of 2003, only 12 testers will be needed instead of 20, saving $295,680. This more than covers the cost of Architect—in fact, CGI estimates the

"It delivers more than we expected."

ROI on the tool at over 700% after just the first iteration. As David put it, "We gained productivity for sure—major gains."(For a breakdown of time savings in the development process, see the Appendix.)



- The graphic presented is an adaptation of the Rational Unified Process (RUP) which is commonly used in large organizations.
- Using Codagen Architect reduces the effort associated with almost all lifecycle activities.
- Codagen Architect also increases the quality of the resulting application by promoting reuse, thus ensuring quality of code and reducing software defects (bugs).
- Codagen Architect allows for much better maintenance and evolution of an application by letting you introduce new technologies/versions (DB, app server, etc.) and propagate the required glue code throughout the application.
- This means Time + Cost savings of about 30%.

Codagen™
The evolution of development

for more information, visit
www.codagen.com
or call 1.877.codagen

Codagen Architect in your development process

Process Time with and without Codagen Architect

	Without Architect	With Architect	Savings
Business Modeling			
Capture business vision	20	20	0%
Define business rules	50	50	0%
Define scope	30	30	0%
Develop business use case model	75	75	0%
Total	**175**	**175**	**0%**
Requirements Management			
Understand stakeholder needs	150	150	0%
Analyze the problem	200	200	0%
Define the system	250	250	0%
Manage the scope	100	100	0%
Refine the system definition	150	150	0%
Manage change	50	50	0%
Total	**900**	**900**	**0%**
Analysis and Design			
Define a functional architecture of the system	200	200	0%
Detail the analysis packages	100	100	0%
Realize the use cases	150	150	0%
Detail the analysis classes	100	100	0%
Analyze the data needs	50	50	0%
Refine the system architecture	200	50	75%
Detail the subsystems and their interfaces	300	100	67%
Realize the interfaces of the subsystems	200	50	75%
Finalize the design classes	50	25	50%
Define the physical data model	100	100	0%
Total	**1450**	**925**	**36%**
Implementation			
Define the implementation model	20	20	0%
Define the integration plan	20	20	0%
Carry out template implementation and code generation	0	50	
Integrate each subsystem	100	50	50%
Develop specific code and execute unit test	400	150	63%
Total	**540**	**290**	**46%**
Test and QA			
Develop a plan for functional tests	30	30	0%
Design functional tests	150	150	0%
Execute functional tests	350	200	43%
Total	**530**	**380**	**28%**
TOTAL	**3595**	**2670**	**26%**