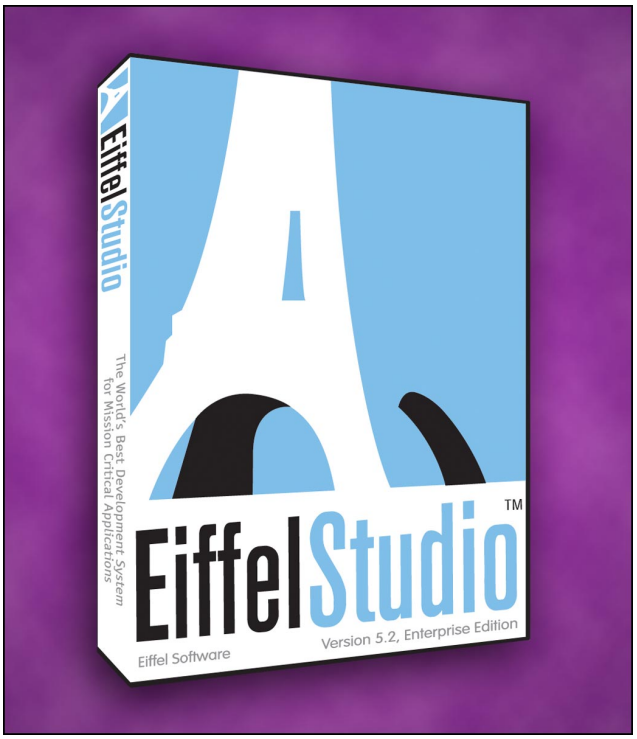
Eiffel Software

C-Level Executives Discovering That Choice of Development Software Drastically Affects the Bottom Line

Increasing acceptance of the EiffelStudio development system proves that software can reduce costs, speed time-to-market and improve product reliability while fulfilling company goals.

Business leaders of late have been changing their vision of what they see when peering into the monitors of their computers. No longer do they view software as merely a necessary adjunct to surviving in the technology era, but, rather, a gateway to producing bottom-line results for the organization. Much of this is due to the latest generation of application development tools that enable companies to develop their ideas rather than their implementations—addressing real world ***business*** problems rather than solving ***programming*** problems.

With raised expectations, the smartest business leaders are now immersing themselves far deeper into the software selection process than ever before. No longer satisfied with rubber-stamping business as usual, they are demonstrating why the optimum application development tool can dramatically affect productivity, labor costs, hardware costs, delivery timelines and product robustness. The fact that software selection affects return on investment and helps fulfill an organization's mission statement elevates its importance right into the executive office.



The increasing acceptance of the EiffelStudio development system proves that software can reduce costs, speed time-to-market and improve product reliability while fulfilling company goals.

The need for more efficient software tools

Nearly every major business now finds itself in need of computer-aided software engineering to improve its products and processes. From financial services, food production, healthcare, construction and even service contractors, one emerging object-oriented development framework can reduce software development project costs by several hundred thousand dollars while delivering product to market in half the time. Given customers' demands for increasing functionality, shorter delivery schedules and long term support—all at a competitive price—there are no excuses for failing to consider these new tools.

"The software industry is in a long, painful transition from process-oriented software development to

object-oriented software development," observes Rex Fowler, founder and CEO of Fowler Software Design LLC of Denver, Colorado—which develops business software for a large number of diverse industries throughout the country. "There is simply no alternative to this trend. Designing software modules around 'objects' such as invoices, employees, customers and purchase orders permits a far better management of complexity than designing software around processes."

Even popular programming languages like C++ can no longer keep up with the latest breed of simplified object-oriented languages, according to Fowler.

"Because C++ had to conform to C as it was developed, it now has too many tricks and traps," says Fowler. "For example, C++ programmers have to manually allocate memory, and then carefully de-allocate memory for all of their applications. If they run into any problems with this allocation, you can have some horrendous fatal errors. And modeling a system that will be built in C++ also requires the use of a universal modeling language product like Rational Rose, which can get very complex."

In the case of most modeling languages, the word "complex" is read as meaning expensive to use, often costing $250,000 for a typical two-year project with annual maintenance fees approaching $100,000, by some estimates.

"We used to use C++, but in 1997 we switched to Eiffel because it is the best object-oriented language on the market today, even better than Java," adds Fowler. "By using EiffelStudio we save as much as half, sometimes more, in development and maintenance costs because you don't need extra modules to address the entire development cycle."

The latest "power tool"

Founded in 1985, Eiffel Software is a leader in providing "true" object-oriented programming tools and component libraries for business-critical and enterprise software developments. The Eiffel language serves as the foundation for the company's EiffelStudio™, a true object-oriented development framework that is not based on C, C++, Java, or C#. EiffelStudio is used in place of more costly and complex programming environments and tool combinations such as Forte/Sun One, Visual Studio, VS.NET, Eclipse, Rational Rose, Togethersoft, Delphi, and several others.

The efficiency of Eiffel results from its compression of the development process, which traditionally requires a word processor to draw up a system definition document, then a modeling tool to create functional blocks and interrelationships, a development tool to produce the code, perhaps a debugger, and finally a separate testing tool for extensive testing and optimization. But with an integrated development environment (IDE) like EiffelStudio, the entire process is completed in a single environment and language. For this reason, most add-on tools become unnecessary, resulting in tremendous cost savings for the overall project, in addition to greater ease of use for the developers.

Greater productivity that cuts straight to the bottom line

The use of EiffelStudio can save as much as 65 to 75% in costs compared to Microsoft VisualStudio™ or a Java environment, under a fact-based comparison of programming environments for a typical enterprise application development. When talking about a full e-commerce setup containing about 2,000 active classes and requiring a number of Web, application, and database servers, and a data warehouse, that means instead of a $10 million bill, it's closer to $3 million with Eiffel. The savings in annual maintenance costs are usually even higher.

In the case of one company, a similar scenario turned into reality when a team of only four Eiffel programmers produced a fully functional configuration management system for a new media file server in only five months, without overtime. Compared to other divisions within the company that used other development tools, the Eiffel team produced a time-cost savings of nearly 36 to 1, an improved time to market from 720 man-months per product down to 20 man-months, while improving system reliability to six-sigma levels. That division's success enabled the company to grow the product line from zero to $1 billion in

seven years.

According to some hands-on CEOs, the correct choice of software can yield a rapid return on equity. Rex Fowler achieved his company's objectives by leveraging his software choice to provide a competitive advantage. Since going with Eiffel four years ago, revenue growth has been steady and dramatic every year. Fowler estimates that they take a 30% cost advantage into any competitive bidding situation where Eiffel software is matched toe-to-toe with C++ software.

"We can use Eiffel to come in under-bid and on time, so that we look like a hero," states Fowler.

In a recent interview posted on Artima.com, competing-language Python inventor Guido van Rossum commented on productivity as a function of lines of code.

"I heard someone say that a good programmer can reasonably maintain about 20,000 lines of code," said Rossum. "Whether that is 20,000 lines of assembler, C, or some high-level language doesn't matter. It's still 20,000 lines. If your language requires fewer lines to express the same ideas, you can spend more time on stuff that otherwise would go beyond those 20,000 lines."

However, Eiffel enthusiasts heartily disagree with this logic. They contend that readability and structure have everything to do with how maintainable a system is—which dramatically affects developer and team productivity. Users say that the average Eiffel developer can maintain 100-200,000 lines of code, and the most proficient can maintain up to *one million* lines—about 50 times the industry standard. This level of productivity is extremely hard to approach in any other language.

In selecting EiffelStudio as a recipient of the 10th annual "Crossroads A-List Awards" for 2003, contest sponsor and independent analysts Open Systems Advisors, Inc. commented: *Compared to similar projects, Eiffel teams have consistently developed more code of higher quality with fewer people...with fewer errors and less testing.* The A-List award recognizes the best, newly proven products and services that are driving business results today.

"Without a doubt, the selection of the software language affects program costs significantly," says Darren Hiebert, principal software engineer for XonTech, Inc. Headquartered in Van Nuys, California, XonTech develops science and technology-based solutions that support the critical challenges of national defense through the simulation of missile defense, sensor, space, launch and target systems. "I was the one who recommended the use of the Eiffel language for our simulation project, so convinced was I of its superiority to existing languages."

Reduced labor costs

A development language directly affects labor costs, as observed within a large number of companies with extensive programming projects.

"Given the shortage of developers, I've noticed an increasing trend in the software industry of hiring people with just 2-3 years of experience," observes Hiebert. "So we need languages that recognize these limitations and provide an environment where human mistakes are not very costly. A language like Eiffel can help mitigate errors because a developer can't get too deep into trouble."

Unique to Eiffel, its code is written in plain English syntax and is readable by a variety of users. The intuitive, graphical user interface is easy to learn and facilitates collaboration among project team members, no matter their level of experience. It also seems to have a positive impact on employee turnover.

"It is much easier to train Eiffel programmers than C++ programmers," says Fowler of Fowler Software. "But even more important from my perspective as a manager, is that back in old days when we were doing C and C++ programming, our turnover was horrendous. We'd bring somebody in, teach them C++; they'd stay for a while, but then leave. Then I'd bring somebody else in and the same thing would happen. It was very expensive for our company to operate that way. But with Eiffel, I haven't seen anything close to the previous turnover because the programmers enjoy their work."

Unique to Eiffel is its mechanism for preventing the creation of over 90% of bugs and the associated debugging expense. Known as Design by Contract™, this novel feature ensures that the implemented soft-

ware directly meets the requirements set out in the system design phase. It also makes it extremely easy for anyone to understand what is going on in the entire system. In terms a business executive can appreciate, Design by Contract introduces the business concept of accountability into programming.

EiffelStudio also extracts hyperlinked html-based documentation from the latest version of the code itself, automatically generating a real-time picture of the system that virtually anyone can read. This enables onlookers to better understand the global meaning of a program without looking at the core implementation. EiffelStudio's auto-documentation feature gives the company a constantly-updated document that makes it much easier for both the people who wrote the code, as well as any newcomers to the team, to understand and maintain the system. Eiffel's users report that this reduces 80-90% of the cost of software system maintenance, essentially freeing development resources for other tasks.

"With Eiffel I can do more with less, but I would rather grow our revenue rather than shrink our staff," comments Fowler. "Between January 2000 and March 2002, we tripled our revenues."

Reduced hardware and licensing expenses

Software should be flexible enough to operate on a variety of existing or proposed platforms to allow company buyers leeway in selecting the most cost-effective platforms. The EiffelStudio environment is almost certainly the most multi-platform flexible "O-O" development environment, running on Windows (NT, 2000, XP), Linux, Unix (Solaris), SunOS, HP 9000, IBM AIX, Unixware, Silicon Graphics, Data General, Fujitsu, DEC, VMS (Alpha and VAX). A Mac OS X version is presently in beta testing.

"Eiffel helped us migrate our business system off the large and more expensive DEC, VAX platform onto Windows 2000™ and XP," comments Dr. Mark Howard, Director of Software Engineering for Barr Rosenberg Research Center, the research arm of AXA Rosenberg Investment Management LLC. The Orinda, California-based firm is a recognized leader in the systematic analysis and management of equity portfolios. "Our systems are now virtually platform independent. The immediate advantage is that we are now running our system on much cheaper hardware, but more importantly we are no longer making an implicit bet on technology for the survival of our knowledge base."

"Since we use EiffelStudio, we are now contemplating switching over to Linux, which is a less expensive platform because you can buy stock PCs and run it on them," agrees XonTech's Hiebert.

To reap maximum savings, any switch in programming software must also allow an organization to maintain its investment in existing information. For example, Eiffel is an open system that can interface with many languages to preserve the best results of earlier efforts.

"Eiffel allows you to consolidate legacy code and company knowledge into a central, more functional and accessible application," notes AXA Rosenberg's Howard. "What you do is you write Eiffel classes for the things that you still need the legacy stuff to perform. The Eiffel "wrapping" becomes your representative for those legacy things, making them objects."

Bringing products to market sooner

New generation programming software also helps release projects on schedule, helping to provide a competitive edge at best, and avoid expensive penalties for late deliveries in a worst-case scenario.

"We can produce systems 3-4 times more efficiently and quickly with Eiffel than we could with C or Fortran," says Howard. "There is an up front cost in actually representing what you know in any computer language but Eiffel allows us to evolve our systems much more efficiently."

"We are continually enhancing one existing product for use in the defense industry," adds XonTech's Hiebert. "We make goals for ourselves as to what capabilities we will have in our next release, and it is pretty much always on time. I've worked with C++, and I'd be willing to estimate that it would take at least two times longer to develop it in that language as opposed to Eiffel."

Software that works for the future of the organization

"Our employees now have a better, faster way to learn how to best contribute to the company," observes Howard. "Our new ability to share knowledge shortens the learning curve by 50%, and takes employees 50% further in their understanding of our business. We've been using Eiffel for ten years here and every new thing that we put into it reinforces the feeling that doing business any other way would have been misguided."

"Our industry has changed over time; development is now a much different activity than it used to be," continues Howard. "Using something like EiffelStudio enables us to develop our ideas, rather than our implementations. What ends up happening is you can address *business* problems, not *programming* problems. It gets right to the point on what the company strives to do, allowing it to better fulfill its mission."

For more information about EiffelStudio, contact Eiffel Software at 356 Storke Road; Goleta, California 93117; (805) 685-1006; sales@eiffel.com; or visit www.eiffel.com.