



AVOIDING THE EPIC FAIL

IMPROVING SOFTWARE DEVELOPMENT
SUCCESS WITH ACTIVEPERL

ActiveState

AVOIDING THE EPIC FAIL



WHY IS SOFTWARE DEVELOPMENT SO HARD?

It is easy to sympathize with software developers trying to build large, complex enterprise software solutions. At the start, a software development project is like a smooth sheet of ice: full of possibility. It's a clean slate, free from architectural flaws, bugs, and broken code. But once the first line of code is written, complexities begin to layer one on top of the other. Before you know it, your software project looks more like the aftermath of a hockey game than a pristine ice surface.

Software development is hard. So hard that industry studies, like the 2006 Chaos Report from the Standish Group, document failure rates as high as 65%. No matter what the end product, many of the same stumbling blocks come up time and again and get in the way of project success. Some classic pitfalls include time restraints, insufficient in-house resources, scope creep, and spiraling development costs. So how do software developers minimize these obstacles and improve their chances for better, faster, more cost-effective software development?

In this paper, we present the open source Perl programming language as a solution for avoiding typical software development foibles. We discuss how an enterprise grade version of Perl can cure the headaches that go hand-in-hand with building and managing specific aspects of your software development project in-house. By harnessing the processing power of Perl and implementing a fail-safe Perl distribution with enterprise-grade support, developers can get quality software products out the door faster and cheaper.

COMMON SOFTWARE DEVELOPMENT PITFALLS

There are plenty of land mines in software development. These are some of the obstacles that can delay your development project:

When a software development project falls behind, the business starts to lose money, and the project falters.

- 1. An Unrealistic Development Schedule** | Software development projects are a race against the clock. All too often, there is not enough time to build the product's core feature set, let alone satisfy new requirements that inevitably make their way into the design spec. When a software development project falls behind, the business starts to lose money, and the project falters.
- 2. Resource Scarcity** | Even if you have a software development "Dream Team" in-house, you are bound to come up short. That's why there has been an explosion in third-party software and outsourced development over the last decade. Organizations just don't have all the resources to manage every aspect of their development projects. Not recognizing your company's resource limitations can be a software development project's undoing; re-inventing the wheel in-house puts your project behind schedule, reduces your competitive advantage, and puts a strain on the business.

AVOIDING THE EPIC FAIL



- 3. Underestimating a Project's Complexity** | Software development presents hundreds of technical issues to solve. From crossplatform support to performance and scalability, each technical challenge puts stress on an over-subscribed development team that often did not anticipate the number of technical challenges they would encounter. As a development project unfolds, the technical issues amplify and become unwieldy, overwhelming, and expensive to fix.
- 4. Being Held Hostage by Third-Party Vendors** | Using third-party software components can help reduce the complexity of your project and can even improve its quality. But proprietary third-party software comes with its own set of challenges. Most critically, your project can get sidelined while third-party vendors take their sweet time making fixes or adding new features. Putting your project schedule in the hands of another software vendor can threaten your success.
- 5. Not Budgeting for Developer Expertise** | The right engineers will always create better software than average programmers. Invest in quality people—whether they're in-house or external experts—because getting your hands on the right development expertise is your first defense against project failure.
- 6. Budget Restraints** | You may have already noticed, but there is never quite enough money budgeted for software development projects!
- 7. Dismissing Open Source** | The popular tech journal InfoWorld said, "The most conservative IT shops dismiss open source solutions as a matter of policy [and] that's a big mistake."

PERL TO THE RESCUE: AN OPEN SOURCE SOLUTION THAT IMPROVES YOUR CHANCES OF SOFTWARE SUCCESS

So, how does Perl improve your chances of successfully completing your software development project? For starters, it's a proven solution used by millions of software developers around the world for a variety of programming tasks. Why reinvent the wheel in-house? It takes resources away from building core features and burdens an already oversubscribed development team. Perl is immediately available and, because it's open source software, there are no licensing fees if you choose to use a free, unsupported version of Perl. Plus, the ever-improving, extensible nature of open source languages means the code is constantly adapting, evolving, and stabilizing. There is a colossal community of Perl developers continuously working to improve and enhance Perl with upgrades and fixes. Like any platform that is widely used, Perl progresses over time, adding to the stability and scalability of any software product.

What is Perl?

Perl stands for Practical Extraction and Report Language and is a general-purpose dynamic programming language, originally designed to make report processing easier. It's sometimes called the "duct tape of the Internet" because it's so flexible and adaptable. While manipulating text files is its specialty, Perl is used for a variety of computational tasks: system administration, network programming, CGI pro-

By rejecting open source you miss out on proven, stable, and scalable low-cost solutions.

AVOIDING THE EPIC FAIL



gramming on the web, and in applications with a back-end database. This description of Perl from Wikipedia is rather charming: “While other programming languages—such as Fortran and C—were designed to make efficient use of expensive computer hardware, Perl is designed to make efficient use of expensive computer programmers.” For this reason, Perl helps to keep software development projects on schedule and on budget. By rejecting open source, you miss out on proven, stable, and scalable low-cost solutions.

Perl is popular with system administrators and CGI script authors because it does such a fine job of processing large amounts of data for creating reports or running test cases. It’s a favorite for database applications because it handles SQL queries gracefully and efficiently. Perl also scores high marks for interoperability and cross-platform development because it painlessly unifies systems and interfaces that were not designed to play nicely together. Perl is widely used by high-traffic websites because its database-caching feature helps manage intense traffic spikes; websites including [bbc.co.uk](#), [Amazon.com](#), [LiveJournal.com](#), [Ticketmaster.com](#), and [IMDb.com](#) use Perl for precisely this reason.

The Perl Business Boost

Perl makes business sense. As an open source language, you’re not required to pay license fees. Not having to shell out for license fees can reduce the cost of software development and help keep

your project on budget. When your team is under pressure to deliver with few resources and a tight budget, the Perl developer community can lend a hand by making fixes (albeit on their own schedule) and providing support where needed (if they are so inclined). Plus, Perl’s agility and continuous upgrades and fixes give software projects deploying open source software a competitive advantage.

But, Perl is not a panacea. There are some disadvantages that come with incorporating open source into your software development project. For instance:

- › 24/7 support doesn’t exist. When you are on a tight development schedule, a lack of formal support or training can put your project at risk. The open source community will respond but, like third-party vendors, they work on their own schedules.
- › If the open source software you’re using is not top quality, it can cause a ripple effect that disrupts your product quality and/or business processes.
- › Managing open source licensing can be both an administrative headache and opens your business up to legal hassles. If you misinterpret licensing requirements, you could wind up in the middle of a legal battle. The last thing you want is to see is all that money saved on license fees redirected to lawyers’ fees.

Perl is popular with system administrators and CGI script authors because it does such a fine job of processing large amounts of data for creating reports or running test cases.

AVOIDING THE EPIC FAIL



TAKING A HARD LOOK AT PERL: THE PROS AND CONS

PROS	CONS
Perl is “mature” It’s been around for a long time and is freely available	Its syntax can be difficult to learn and can cause usability issues
Good for file manipulation	No 24/7 technical support available
Often used for CGI scripts and other web-related jobs	Perl scripts can be difficult to write and maintain
It’s commonly used, especially by system administrators	Open source licensing can be an administrative and legal hassle
It’s portable: available for almost every platform	
It’s good at pattern matching, regular expressions, and string manipulation	
Recognized for fast processing speed	

Perl Fun Facts:

- *Perl celebrates its 21st birthday this year.*
- *Perl was originally named “Pearl” after the “Parable of the Pearl” that espouses the value of heaven, from the Gospel of Matthew.*
- *Programming Perl, published by O’Reilly Media, features a picture of a camel on the cover. The camel image has become a symbol of Perl.*
- *Perl slogan: “Easy things should be easy and hard things should be possible.”*

ACTIVEPERL: FIRE AND FORGET PERL

Harnessing the power of Perl means software developers don’t have to cobble together all their own solutions for data processing, CGI scripting, and system administration, but the risk is that developers spend the time saved in implementing Perl solutions on rebuilding Perl installations and managing all the updates and fixes along the way. If Perl isn’t an area of in-house expertise, maintenance can be daunting and time consuming, so it won’t mitigate schedule and budget creep.

ActiveState’s ActivePerl Enterprise is a no-risk Perl solution for developers who want to integrate Perl into their software projects, but don’t want the hassle of managing and maintaining it. ActivePerl is ‘fire and forget’ Perl. It’s a ready-to-install, industrystandard Perl distribution that works across a variety of platforms: Windows, Linux, Mac OS X, and others. It is used by millions of developers around the world for easy Perl installation, quality-

AVOIDING THE EPIC FAIL



assured code, and comprehensive technical support. While it is entirely possible to compile Perl source code from the Comprehensive Perl Archive Network (CPAN) website—where Perl modules, scripts, distributions, and source code are available for free download—many systems don't include the requisite C compiler and build tools. As such, installing modules can be tricky and potentially time consuming.

ActivePerl comes pre-compiled for out-of-the-box installation and includes core Perl, popular modules, the Perl Package Manager (PPM), and complete documentation. The Windows version provides additional features that have made ActivePerl the worldwide standard for Perl on Windows. ActivePerl Enterprise Edition software is packaged with commercial-grade documentation, technical support, and access to some of the industry's most knowledgeable Perl experts to help with installation, upgrades, and maintenance.

Maintenance-Free Perl Cuts Down on Project Complexity

You can lessen project complexity and resource scarcity by handing maintenance over to the Perl professionals. ActivePerl Enterprise Edition customers have been known to save days—even months—of developer time by eliminating the work it takes to build, test, and maintain their own open source distributions.

Open Source Expertise at Your Fingertips

Adding a Perl expert to your development team can make the difference between good and great software. Rather than hire in-house, which can be expensive and possibly even excessive, you can lean on ActivePerl language experts for design and development advice. ActiveState engineers contribute heavily to Perl and maintain great relationships with

the open source community. They continually and rigorously quality test the ActivePerl distribution to ensure that the code boosts the overall quality of any commercial software project.

Cost Effective Licensing

At first glance, managing open-source licensing on your own seems the most affordable option. But open source licensing can be baffling. Once you point a developer or two at collecting and reviewing all the licenses

in use and then get lawyers onboard to do due diligence, the cost of managing Perl licensing in-house can add up. ActivePerl OEM Licensing provides an easy, out-of-the-box, fixed licensing solution that guarantees you've got the right licenses in place. It takes open source licensing issues off your plate and reduces the roadblocks to software delivery.

ActivePerl = Shorter, Cheaper Development Cycles

ActivePerl typically helps to shorten development cycles and increase developer efficiency. Numara is a development shop that has used ActivePerl since 2005. As Numara Vice President of Product Development said: "We came to the realization that there was no point dedicating our energy and resources towards rebuilding Perl when ActiveState could do it more cost effectively."

"We came to the realization that there was no point dedicating our energy and resources towards rebuilding Perl when ActiveState could do it more cost effectively."

AVOIDING THE EPIC FAIL



CASE STUDY: CA CHOOSES ACTIVEPERL FOR SMOOTHER SOFTWARE DEVELOPMENT

CA is one of the largest independent software companies in the world. CA SiteMinder Web Access Manager—one of CA's most popular products—has more than 1400 customers, including the majority of the Fortune 500. CA SiteMinder is an enterprise-scale Internet access control system. It provides secure access to web applications and websites for employees, customers, and business partners.

CA SiteMinder chose Perl for an API that runs test automations. Perl was a natural fit for the task because it's well suited to parsing and manipulating large amounts of data. It's also a familiar and simpler programming language for CA's customer base of system administrators, database administrators and IT professionals than either Java or C++. Plus, Perl scores high in interoperability and expedites cross-platform development, an important feature since SiteMinder runs on more than 20 different platforms.

But, when CA investigated Perl in 2000, they weren't convinced the existing free version was stable enough for their needs. Lawrence Backman, a Vice President at CA, says at the time, open source Perl had some multi-threading and multi-processor issues as well as scalability problems: "CA SiteMinder is an enterprise-level product, so we had to be sure we were using a solid, trusted version of Perl," said Backman. So, in 2001, CA integrated ActivePerl into SiteMinder and other tools in CA's web access control product family.

SiteMinder now uses ActivePerl to run test automations for the SiteMinder family of products. The product set spans a test matrix of more than 350 distinct platform

combinations. Because ActivePerl supports all these platform combinations, CA has been able to take Perl to 32/64 bit platforms of all the major operating systems without having to worry about OS or platform differences. These test automations are the first line of defense against inadvertent regression—instances where a feature stops working or performance slows.

"ActiveState has both allowed us to continuously raise the bar on quality as well as cut the cost of expanding our product matrix."

LAWRENCE BACKMAN

"Using ActivePerl, we are typically able to find and address a regression in the core product within 24 hours of inadvertent defect insertion," said Backman. CA does regular maintenance on SiteMinder, which may result in anywhere from 25 to 30 fixes a month. It is easy to unintentionally introduce regressions during a fix; automated testing ensures that does not happen. If it does, CA fixes the problem immediately.

"ActiveState has both allowed us to continuously raise the bar on quality as well as cut the cost of expanding our product matrix," said Backman. Although CA's development team has the know-how and experience to successfully manage Perl code that is available free from the

"CA SiteMinder is an enterprise-level product, so we had to be sure we were using a solid, trusted version of Perl."

LAWRENCE BACKMAN

AVOIDING THE EPIC FAIL



CPAN website, Backman says using ActiveState's enterprise-level distribution makes more sense. Not only is it reliable, scalable, and enterprise-ready, ActivePerl saves CA development time and, therefore, dollars. Maintaining, updating, reinstalling, configuring and porting Perl is just not the best use of CA's developers' efforts. By leaving Perl infrastructure maintenance up to ActiveState, CA's developers can focus squarely on enhancing SiteMinder and developing new features that drive the product's marketability and revenue, says Backman.

IS ACTIVEPERL RIGHT FOR YOUR PROJECT?

At the start of every software development project, there is hope and anticipation for smooth, problem-free development. But before your clean sheet of ice gets ruined, put a plan in place to help eliminate typical software development downers: time restraints, insufficient in-house resources, and spiraling development costs. Integrating Perl into your enterprise software development project can help keep you on schedule and on budget. By choosing an enterprise distribution of Perl, like ActivePerl, you ensure top quality code, do not have to worry about maintenance or upgrades, and add Perl expertise to your development "Dream Team" without having to hire in-house.

Before your software development project starts down a path towards complexity and chaos, talk to an ActiveState open source expert to determine if Perl is the right fit for your software development project. ActiveState experts will weigh and consider all the challenges your development team faces and make recommendations tailored specifically for your project. Register today for a complimentary consultation with an ActivePerl specialist.

Contact ActiveState at 778.786.1101, or business-solutions@activestate.com for a complimentary consultation with ActiveState's language experts.



ActiveState Software Inc.

sales@activestate.com

Phone: **+1.778.786.1100**

Fax: **+1.778.786.1133**

Toll-free in North America:

1.866.631.4581

ActiveState[®]

ABOUT ACTIVESTATE

ActiveState believes that enterprises gain a competitive advantage when they are able to quickly create, deploy and efficiently manage software solutions that immediately create business value, but they face many challenges that prevent them from doing so. The company is uniquely positioned to help address these challenges through our experience with enterprises, people and technology. ActiveState is proven for the enterprise: more than two million developers and 97 percent of Fortune 1000 companies use ActiveState's end-to-end solutions to develop, distribute, and manage their software applications written in Java, Perl, Python, Node.js, PHP, Tcl and other dynamic languages. Global customers like Cisco, CA, HP, Bank of America, Siemens and Lockheed Martin trust ActiveState to save time, save money, minimize risk, ensure compliance and reduce time to market.