

## Use Coding Standards to Increase Efficiency and Productivity

## **INTRODUCTION**

As companies work to overcome the economic downturn, managers and executives are further scrutinizing IT purchases, leaving many of us with the job of building a complex business case to justify the technology we need. Purchases are often evaluated on a simple ROI based on the length of time necessary for the cost savings achieved to offset the cost of the purchase. This provides a limited view of the solution's value, and can lead to the exclusion of many important factors. For example, the implementation of coding standards is so closely linked to other aspects of software development and business process that it is difficult, if not impossible, to evaluate the implementation based on a simple price to savings ratio. Instead, we recommend that organizations evaluate not only immediate cost savings, but also overall productivity gains.

The purpose of this paper is to provide you with information on the importance of coding standards, and how using them effectively can help your organization achieve a variety of benefits, including reduced development costs, and increased efficiency and productivity.

## **WHAT ARE CODING STANDARDS?**

Before we get into the details of the business case, let's define what we mean by coding standards. Many software industry professionals view coding standards as a way to "beautify" code, or to apply simple, consistent patterns that check naming standards (such as requirements for database field ID names). In contrast, we view coding standards as comprehensive, language-specific rules that describe how software code should be written. Coding standards have been developed by industry experts in almost any programming language to help developers write code that is robust and resistant to the most common and damaging errors.

## **BENEFITS OF CODING STANDARDS**

There are many benefits to using standards in any manufacturing industry. Whether a company produces automobiles, electronics, or computer hardware, standardization of processes and practices can increase efficiency, improve productivity, and reduce costs. Software is no different: by defining and using good coding practices, your organization can achieve the same financial and organizational benefits.

### **Standards increase efficiency**

According to industry experts, software developers spend approximately forty percent (Basili/Boehm 01)<sup>i</sup> of project time fixing issues that could have been resolved earlier, or could have been prevented entirely if the right measures were in place. In a standard workweek, this means that each of your developers could be spending two full days on unnecessary or redundant work, leaving only three days to complete tasks associated with moving a project forward.

By using standards to detect problems earlier or prevent them entirely, your team can increase efficiency throughout the software development process. As a result, the team can be more productive and deliver projects at a lower cost. Analysts at Gartner refer to cost savings from such efficiency gains as Return on Assets, or ROA, because the process and infrastructure put in place allows for savings in time and cost, but also enables an organization to produce more with existing resources. As IT and software projects become more intertwined with business needs, the direct financial benefits produced by each individual project are difficult to isolate from those produced by other projects. A true analysis of any project's value requires looking at investments from the perspective of ROA – considering, for example, how much more value can be gained from existing resources once a new technology or initiative is in place.

### **Standards reduce costs and the risk of project failure**

It is well known that a large percentage (industry estimates place the amount anywhere between thirty-one and seventy percent) of IT projects fail or are cancelled<sup>ii</sup>, with much of the cause attributed to software development problems. Coding standards can reduce the costs and problems that often lead to these failures and cancellations.

According to InformationWeek, programmers inject “about one error for every 10 lines of code they write.”<sup>iii</sup> Without a process for analyzing every developer's code and ensuring that it is constructed correctly, you rely on individual developers to write code that is error-free. Seasoned developers may use good coding practices and have the best intentions. However, even the most experienced developers may introduce damaging errors into the code base, especially under conditions where the development process has been compressed and they are rushing to meet project deadlines.

The cost of fixing each error introduced depends upon how soon that error is discovered. Studies have shown that if an error is discovered after delivery/release, it can be anywhere from five to 100 times more expensive than it would be if that same error were discovered during development (Basili/Boehm 01). For this reason, it is critical to ensure that the entire development process takes place in an environment that is hostile to errors. Any poorly written code should be detected immediately, when it is easiest and least costly to fix. Without preventative measures in place, many projects will result in high costs to detect, isolate, rework, and retest code. This can lead to exponential increases in cost that result cancellation of the project.

By simply using coding standards and a formal code review process, your team can catch 60 percent of the defects in code (Basili/Boehm 01), and can do so early in the development process.

### **Standards reduce complexity**

Without standards in place, the code for a single software component might be written a great deal of different ways – some of them very simple and others very complex. One of the generally accepted concepts in software development is that the more complex a piece of code becomes, the greater the probability that it will contain errors—just as the more complex a development organization becomes, the more prone it is to breakdowns in process, inefficient use of resources, and an increase in costs to address any problems that arise. By using coding standards to define how software code should be written, you can ensure that the code your team develops is free of unnecessary complexity, and that the development process does not lead to additional costs in the future. In addition, you reduce the risk of problems arising during the integration of components developed by different companies, groups, or team members.

### **Standards prevent hidden costs**

Whether you're developing software for internal use or for commercial sale, sub-standard quality can cause your organization to incur additional development costs as you fix errors, redesign the software, provide technical support, refund customer purchases, and lose brand value. While few of these consequences are considered during project planning cycles, they can lead to project delays or cancellations, budget overruns, and losses in market share and revenue. Such factors cannot be quantified for the purpose of ROI, but can be equally important-- if not more important-- when compared to short-term savings generated by an IT purchase.

### **Standards enable code reuse**

Coding standards enable your developers to write cleaner, more robust code, and create individual units of code that can be reused in other modules or in future projects. Code that is written according to industry standards is easier to port to other platforms or technologies. For example, as technologies and customer requirements change, you may wish to extend your product to support another operating system, or may decide to extend existing components to offer a new product or service. If the code for those components was not developed using standard coding practices, it could require too much rework to make the opportunity worthwhile.

By enforcing coding standards on every project and every piece of code created, you can ensure that the code is high quality and easy to reuse and extend for other projects. This enables you to invest intelligently in all of your software projects and reuse technology to respond to market demands – giving you a competitive edge.

### **Standards enable staffing flexibility**

Whether you have varied levels of experience within your development team, or have outsourced projects or contract employees, it can be difficult to ensure that code for

each project meets the same quality expectations that you have in place for seasoned programmers.

When all developers working on your code follow coding standards, code is easier to understand, modify, and maintain, whether the developers remain on the same projects over time or move on to other projects.

Moreover, if you choose to outsource, standards enable you to maintain a minimum level of quality for code developed by the outsourcing organization, removing the fear of unknown or inconsistent quality from such projects. All of these factors contribute to a more flexible, agile organization.

### **Standards enable Automated Error Prevention**

Finally, the most compelling reason to implement coding standards is that they enable Automated Error Prevention. AEP is a concept that advocates that each time an error is discovered in software, the process is improved to prevent the error – and all classes of errors like it – from reoccurring. Automated Error Prevention promotes five simple steps, which should be automated whenever possible:

1. Detect an error
2. Isolate the cause of the error
3. Locate the point in the process that created the error
4. Implement practices to prevent the error from reoccurring
5. Monitor for improvements

Coding standards are an excellent example of an AEP practice because they enable you to implement requirements that prevent the most common and damaging errors in software code. Coding standards enable you to implement a technology infrastructure that automatically analyzes code, detects and isolates errors, and supports continuous process improvement.

With defined standards in place, you can use an automated system to analyze code at the desktop level as each programmer writes code; literally detecting the errors as the code is written. You can also enforce code quality across all members of your development team. By requiring that all code checked into your source control system complies with designated coding standard rules, you ensure that no team member introduces errors.

By using coding standards, you can not only detect and isolate errors at the code level, but also prevent later errors that could lead to performance problems or downtime, causing additional, unplanned costs. Standards reduce the amount of testing and rework required later, and your code review cycle becomes a forum for discussing coding standards violations, rather than spending time manually reviewing each piece of software code. When team members violate a standard, the code review becomes an opportunity for them to discuss why the requirements should or shouldn't be adhered to, or to modify the set of standards to better suit your organization's needs.

In addition, with coding standards that are built in to an AEP infrastructure, you can skip the process of creating your own standards, or the AEP requirement to first find an error before improving the process to prevent it. With a solution that can be customized, you can apply only the standards you deem critical and add new rules as necessary over time.

You can adapt to new standards defined by industry groups or your own team members, adding rules individually or upgrading your automated system to incorporate additional rule groups. This enables you to quickly and easily adapt to new technology needs and continue to prevent errors. For example, industry and government initiatives have led to the creation of accessibility standards to ensure that technology is accessible to all users, including those with physical disabilities. Because of how these standards were defined, they also improve ease of use for individuals accessing the technology using handheld or wireless devices. With a customizable system, you can quickly adapt to new issues and prevent errors that accompany them.

## **VI. CONCLUSION**

The implementation and use of coding standards can enable companies to deliver higher quality software in a manner that is cost-effective and efficient. Using coding standards, you can reduce costs, and improve productivity; however, most importantly, you can automate the prevention of errors throughout the software lifecycle. This not only leads to direct savings in time and cost, but also leads to overall productivity gains that will impact revenue and profitability.

---

<sup>i</sup> Basili, Vic, & Barry Boehm. "Software Defect Reduction Top 10 List," Software Management Magazine, January 2001

<sup>ii</sup> The Standish Group, "The CHAOS Report" 1994  
[http://www.standishgroup.com/sample\\_research/chaos\\_1994\\_1.php](http://www.standishgroup.com/sample_research/chaos_1994_1.php)

<sup>iii</sup> "The State of Software" InformationWeek, May 21, 2001