Diving Into Code Quality:
Factors Affecting Code Quality

By Elizabeth So, formerly of DisputeSoft

Businesses across all industries hire software development companies to implement information-technology systems. However, studies conducted by firms such as The Standish Group International, Inc. have shown that a majority of software projects experience serious technical issues, which often result in project failure. One variable that may contribute to the success or failure of a software project is code quality.

Code quality refers to a combination of non-functional attributes of software code, including maintainability, reliability, and security, which together indicate how well code is written. Developers should strive to maintain high code quality across the duration of a software project such that their code adheres to project requirements and technical specifications and meets industry standards for code quality.

Benefits of high code quality include increased maintainability of code, reduction of time spent correcting defects in earlier versions of code, and prevention of critical problems (e.g., medical device malfunction). However, project managers and stakeholders may deprioritize maintaining high code quality during a project’s software development lifecycle (SDLC) to achieve competing goals, such as adherence to a project timeline or staying within budget.

In software failure cases, experts may analyze source code produced during a software project to determine whether poor code quality contributed to project failure. Experts may also review produced documents related to software development processes and project management that provide insight into the factors contributing to poor code quality.

Factors Affecting Code Quality

Factors including skill of the development team, stability of existing software development environments and processes, and strength of project management can affect the code quality of a software project.

Development Team

Developers write code to implement and test functionality that meets project requirements. A team of developers skilled in using the specific programming languages and technologies required for a project generally produces better quality code than a team of developers without a background in those technologies. However, sometimes software development firms hire developers who possess a robust programming background, but who have little experience working with a particular programming language or technology used for a project. For example, a developer may be hired to work on a software project written primarily in language A, even though the developer has written code primarily in language B. As a result, the developer may need to spend time learning about and experimenting with language A before he or she can write high quality code using that language, especially if language A is very different than language B. The amount of time the developer may need to spend learning on the job could result in project delay or poor code quality if the team employs too few other developers skilled in language A to implement the necessary functionality in accordance with the project schedule.

Software Development Environment and Processes

Establishment of a proper software development environment and software development processes enable developers to adhere to coding best practices and more effectively collaborate on code changes, which usually contributes to better code quality. Software development processes beneficial to code quality include:

– Version Control: Version control systems allow developers to review, track, and share code changes. Version control systems also allow developers to maintain code history and link code changes to defects logged in defect tracking systems, which aids in root cause analysis and prevention of recurring defects in the software. In today’s world of complex code development involving multiple developers, use of version control systems has become a standard practice.

– Static Code Analysis: Static code analysis tools perform automated analysis of source code, which developers can use to improve code quality during development. Such tools may catch errors in code and deviations from coding conventions, aiding developers in writing secure and reliable code that is maintainable by current and future developers.

– Manual Code Review: Manual code reviews facilitate discussions of code and design changes prior to feature implementation or defect correction. Developers may perform formal code reviews before code is checked-in to a version control branch in a state ready for deployment, or when investigating results reported by a static code analysis tool. A developer who needs help implementing a difficult feature may request an informal code review from another developer who can provide feedback on his or her code.

– Testing: Developers write tests for software projects to provide quality assurance and to document software's intended functionality. Software projects typically undergo unit, functional, and integration testing prior to release into production. Unit tests assess a small amount of code in isolation; functional tests assess specific functionality in software; and integration tests assess how different pieces of code work together. Running tests during these development cycles helps to identify defects prior to later testing stages (including system testing; end-to-end testing; and user acceptance testing) and to ensure that addition or modification of source code does not break existing functionality. System testing conducted by dedicated internal software testing teams and user acceptance testing conducted by end-users may improve code quality by identifying software defects to be fixed prior to release into production. On the other hand, inadequate testing may contribute to poor code quality.

– Source Code Documentation: Developers include comments to document their source code, which can improve code quality by increasing software maintainability. For example, comments describing how code for a complicated algorithm works help developers who need to modify the code in the future. Developers update comments in well-maintained software whenever they add new code or revise existing code. Additionally, developers may need to follow a specific source code documentation format, depending on the SDLC methodology (e.g., Agile, Waterfall) used. One well known computer science professor commented that coder comments should be written before the code is written. In other words, document the code via coder comments and then write the code in accordance with those comments. Seems like pretty good advice.

Project Management

Clearly defining business requirements for a software project allows developers to write code that correctly implements intended functionality. Unclear or vague requirements lead to unclear design specifications, which can hinder the progress of developers and contribute to poor code quality and incorrectly implemented functionality.

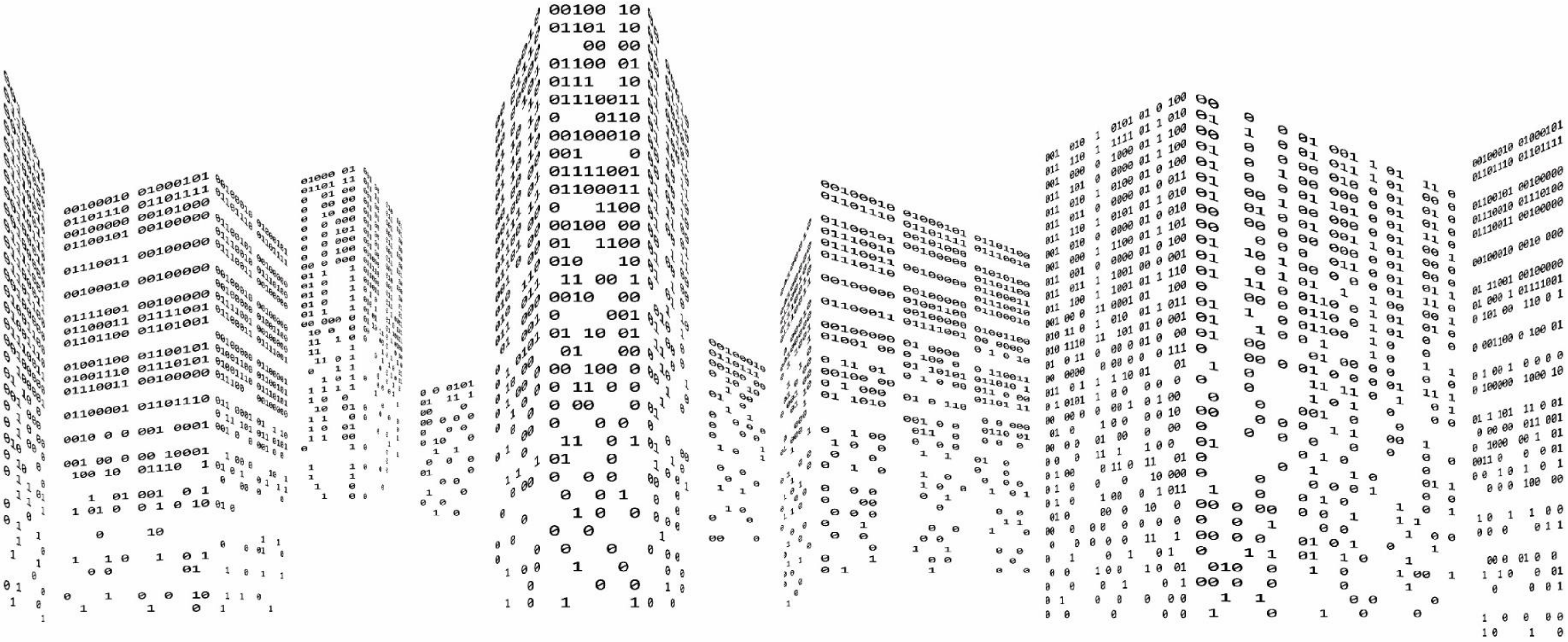
Unrealistic project timelines can also impact code quality. Pressure to complete a project in accordance with an unrealistic schedule or budget may cause developers to create less than ideal software solutions or to rush to implement functionality without proper testing. Both responses can contribute to poor code quality.

Scope creep can also affect a project's timeline. Scope creep refers to addition of new requirements or features which expand a project's scope after the project has begun. Code quality may suffer when project managers and developers lack adequate time to implement functionality not included in the original scope of a project.

Attorneys litigating software failure matters often find that DisputeSoft's analyses concerning code quality significantly inform their understanding of a dispute. To learn about the tools and techniques used to assess code quality, read the second installment in our "Diving into Code Quality" series: "Assessing Code Quality." If you are litigating or evaluating a failed or troubled software project, please don't hesitate to contact us at inquiries@disputesoft.com.

Read about DisputeSoft's software project failure services.

Read about DisputeSoft's source code examination services.



If you are an attorney in need of a source code expert, we invite you to consider DisputeSoft.

Contact Information

Jeff Parmet, Managing Partner

301.251.6182

jparmet@disputesoft.com

12505 Park Potomac Ave. | Suite 475 | Potomac, MD | 20854