Managing Risk in the Software Supply Chain Through Software Code Governance

Kristin Brennan, Coverity

Abstract. With the increasing complexity of software applications, shrinking IT budgets and the spiraling cost of developing software, many organizations in both the public and private sectors are turning to third-party software suppliers including outsourced teams, partners and open source to develop their applications. According to a recent study conducted by Forrester Consulting and Coverity [1], almost all organizations are using some form of third-party code in their products, and over 40% rely on software from three to five different software suppliers.

The use of COTS tools in military environments is no longer limited to hardware. COTS software is increasingly making its way into military platforms. In systems where the use of existing commercial components is both possible and feasible, it is no longer economically feasible for the government to specify, build, and maintain a large array of comparable proprietary products.

However, commercial third-party code is typically not tested with the same level of rigor as internally developed code. As software complexity grows, additional software capabilities bring many more lines of code, and greater opportunity for error. That means a defect could be lurking in the third-party code that could cause a significant breach or security issue.

Organizations are recognizing the need for end-to-end accountability for the quality and security of the code in their products, regardless of who actually created the code. There is a need for efficient processes to enforce consistent software code governance across the software supply chain.

Software Code Governance

The initial focus of software code governance was to assure software quality and security of in-house developed code by establishing clear guidelines and procedures such as the FDA's recommendation that infusion devices be tested with static analysis and DO-178C, Software Considerations in Airborne Systems and Equipment Certification for the avionics industry. Today, we see software code governance gaining momentum in a wide variety of industries as organizations seek to drive greater accountability and efficiency within distributed development teams and to achieve better visibility and control over third-party code.

A Multi-Step Process

Software code governance cannot be achieved with the click of a button. It is a process that needs to be embraced by the organization and enforced across the internal and external supply chain. The process will vary by organization based upon whether you are trying to establish governance across internal teams, with outsourcers, offshore development teams, or partners, and whether you have access to source code for the application.

Establish Acceptance Criteria

Automated code testing solutions enable managers to establish and enforce consistent measures for quality and security across the software supply chain. Organizations can use automated code testing to establish acceptance criteria with their suppliers. For example, it could be mandated in the contract that all code must be tested with static analysis. Static analysis testing produces results that are repeatable, measurable and objective. To support static analysis testing, policies can be automatically established to ensure that there are no uninspected defects and no high impact quality and security defects in the code. A strict acceptance criteria can be established so that all found defects must be addressed before the code is accepted. This approach puts the onus on the software supplier to ensure their code is of high enough quality to pass the established acceptance criteria and would be a practical solution in situations where you do not have access to source code.

Auditing Mode

Another approach to ensuring quality and security across the supply chain is to establish auditing rights with suppliers. Organizations that purchase source code can reserve the right to analyze the supplier's code and report back results. This could be implemented as part of the integration phase of the lifecycle. This auditing right helps the organization measure quality in a consistent manner across their supply chain and with their internal teams. It also enables the organization to provide recommendations and results of the analysis back to the supplier giving them an opportunity to fix the defects. Once a baseline for quality and security has been established with a supplier, a policy can be enforced that no new defects are allowed as new defects could introduce risk into the overall project.

Self-Certification

Organizations who are supplying code can also be encouraged to take proactive measures to "self-certify" the quality of their code before delivering it. NNG, a pioneer of navigation software and the developer of iGO Navigation solutions has adopted such an approach in the private sector. It has deployed static analysis to deliver high quality software and accelerate time-to-market for software delivery to its supply chain. NNG has delivered navigation solutions to more than 150 business customers including the world's leading original equipment manufacturers. Its navigation software is at the heart of millions of products from in vehicle infotainment systems and smartphones to personal navigation devices.

NNG has embedded static analysis into their development process so every new line of code is tested before it is released into the market. It enables them to track and manage defects between 28 projects and different code branches comprising over 1 million lines of code. As a result of development testing, NNG has been able to establish standardized metrics for measuring software quality across the supply chain and remove cost and complexity from its own software development activities.

Conclusion

Establishing and enforcing acceptance criteria and negotiating the right to audit the software quality are two concrete steps organizations can take to maintain the highest levels of quality across their software supply chain. As software and supply chains continue to become more complicated, and organizations continue to deliver more innovation, and at the lowest cost possible, the ability to enforce consistent standards for quality will become increasingly important.

ABOUT THE AUTHOR

Kristin Brennan, Senior Director of Product Marketing, Coverity, has more than 15 years of technology marketing experiences with expertise in development testing, static analysis, code governance, and automation. Prior to joining Coverity, she help senior marketing positions at Hewlett Packard and Wells Fargo Bank. She has a BA from UC Irvine and MBA from UC Berkeley.

Phone: 415-321-5236
E-mail: kbrennan@coverity.com

REFERENCES

1. Forrester Consulting. Software Integrity Risk Report, 2012