University of Missouri – St. Louis

Software Development Security

Security driven development

Sasa Basara
Monday, December 14, 2015

Table of Contents

Abstract

Security design patters in the software development life cycle has taken a back seat to create better user experience. At the cost of this better user experience is the risk associated with the lack of focus in security. This becomes an issue for companies that experience major security breaches and must continue to patch their patterns rather than, taking a security driven approach to designing their software. In this paper we will explore the benefits of moving toward a security driven development pattern. In the paper the focus will be centered on business processes and involving security throughout the organizational ecosystem. The benefits and challenges that is associated with moving toward a security driven development pattern and how to build a security driven organization.

Keywords:

Software Development Life Cycle – This term is used in software engineering to capture the entire process from planning, testing, creating, and deploying the software.

Security requirements specification – The requirements that are based on security needs of the system. These will be different than typical functional and non-functional requirements. They will center on security needs.

Risk management – Managing the risk that is associated with any software product that impacts business process, day-to-day business and overall company health.

System modeling – Using models to represent systems in business and IT development. System modeling typically uses set diagrams to easily show the requirements.

Agile Methodology – A means of developing software that prompts quick delivery of working software that is later iterated.

Security driven development – having development based on of security needs of the application followed by other demands. This includes designing, planning, and requirement writing.

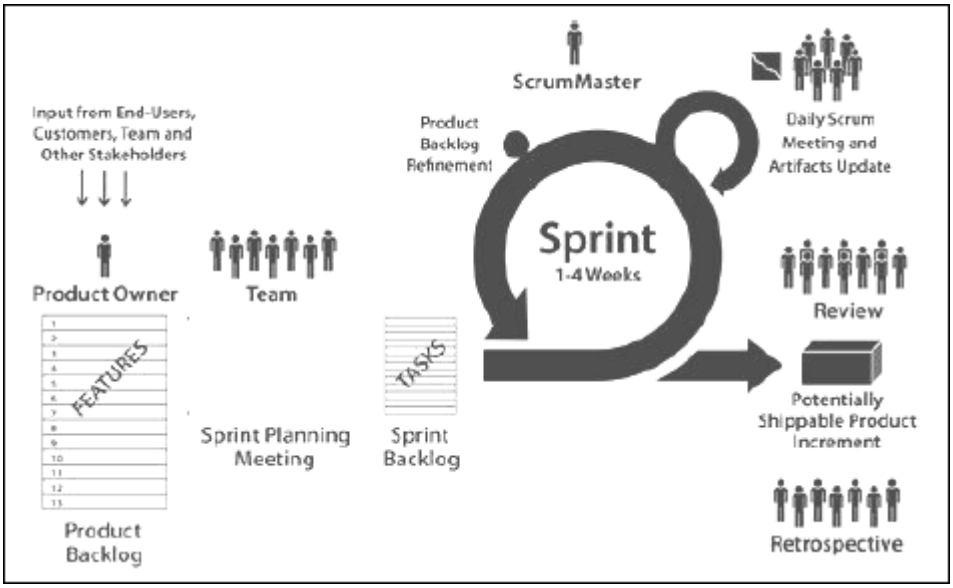
Test Driven Development – software development process that uses automated testing to verify the that code is working to pass a minimum test.

Introduction

When software development started it was a different landscape. There was a concern for security but, not to the demand that is needed today. Throughout the years more focus has been put on having security a focus when developing products However, the industry is in favor of quick iterations, Agile, as they show business progress in a quick fashion than the traditional waterfall approach. Although, this does create better value, it also opens the door for a lot of risk. When developers are focused on assuring that their software hits the market quick they do not factor in the security overhead that each additional change has. With more iteration the risk associated with vulnerabilities increases. These vulnerabilities should be addressed within the agile sprints, but, most of the time they will take a step back to the functionality that is promised to the business.

Background/Historical Analysis

Before the Agile methodology swept over the software development Waterfall was the king. The waterfall methodology focused on building out the entire system requirements before each phase would take place. For example, first the requirements were gathered and then they were analyzed by the development team and the business analyst. Then the development would begin on the product. The product would then go through testing and the back into development if there were any issues. This was a rather costly process if there seemed to be an issue toward the end of the project. However, the benefit of this was that all security was addressed up front and resolved before it reached the next step. Agile's main benefit is that it uses quick iterations. These quick changes allow for costs to be reduced.



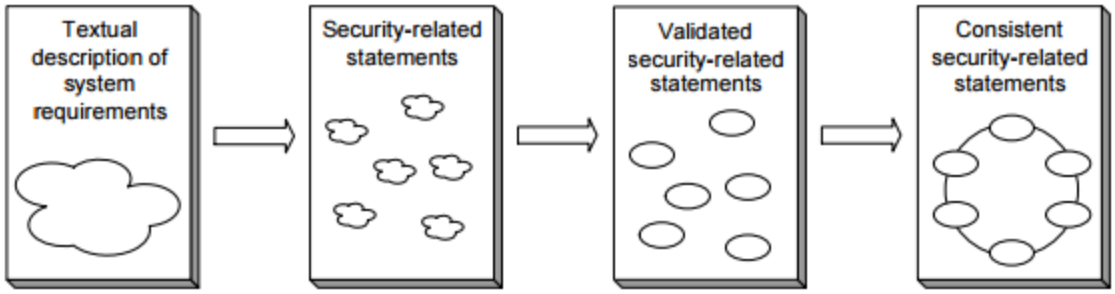
The above figure is provided by a paper on agile projects. Throughout the entire document there is not a single mention of the word security. This is because in the Agile space problems are

handled as they occur. Of course, some major security features are addressed before each release but, the impact that each change cannot always been seen throughout the entire enterprise application. This lack of focus to security is the one of the reasons why there have been countless breaches in major companies. The hackers understand how software development works and understand that certain problems will not addressed until they are put into the teams focus.

Security driven development

With different development methods coming into play as the software development matures. There are many that make more sense than others. A current approach to developing software is Test Drive Development which uses small unit tests to turn out code. Test driven development allows development teams to push out code that meets minimum standards to pass for a functional test. The security driven development, designing a system with security as the main focus, would take characteristics from the current test driven development approach. We must realize that there is never going to be a system that is full protected however, tackling the major risk associated with software is easier to do on the front than waiting for the issues to appear down the road.

Gathering the requirements for a system to be driven by security has to have a formal structure. In general there would be discussion regarding the textual description of system requirements which is then moved to security-related statements and then we would validate those security-related statements. The following diagram would be the flow.



These requirement gathering efforts would help the system developers understand the security risks that are factors. However, security driven development does not stop at the developer's machine.

For this to become a pattern in the development department there needs to be an understand of the pattern that is set forth by the security model. Typically a pattern contains a context, the issue, the forces, and the solution to a particular case. Turning these patterns and requirements into development task takes a flow that touches the entire security development approach. This coupled with the understand of the business furthers the strength of the company that utilize a security driven development approach.

Business Driven Security

To further understand the needs of the business and to share these needs with developers. Having a asset value scale that determines the risk associated with each business logic is beneficial to combining the needs of business and the understand where the two departments can work together to benefit the entire enterprise. A proposed asset values are:

Extreme: Endangering human life or threating enterprise

Very High: Serve financial or security consequences

High: Impact on customer services and reputation

Medium: Affect the enterprises mission

Low: Minor financial damage and little business impact

Negligible: No security relevance

This assets rating system will allow the business to prioritize the needs and to further combine the needs of the business with the capabilities and understand of the developers. The need for this is that typically developers are not as intertwined with the core business as they ought to be. In addition to completing the asses rating system a company will also model their business process to determine where the areas of risk are most likely to occur. These areas are typically in ‘handshake’ exchanges. A handshake exchange is when there is an exchange of information from one part of the system into the other. Another area that would have vulnerabilities would be an external sourcing data.

Issues & Problems

There are two major issues that occur when a company moves toward a new development patter.

Price: The price associated toward moving to any new development model is always high. However, moving toward a development model where security is the focus will incur more cost than the typical switch to a test driven model. The reason is that that with security comes the overhead of learning the security that is behind the code. Most developers do not have a firm understanding of security concepts as they are not typically given focus in academic CS programs. This cost alone would be enough to turn business away toward moving to a security driven development model

Cultural: The greatly overlooked aspect of most engineering companies is the cultural of the developers. For example, a company that is use to pushing out products quickly might not understand that they need to take some time to evaluate their security risk with each deployment. These could cause a tear in the company as more features as built. This risk can be considered greater than the price risk that is associated because; it touches the core of the company’s mission.

Solutions and recommendations

The recommendation would be to advise the companies to take the cost up front rather than when an issue arises. Having a risk management plan and understand the assets that are strongly tied to business development will likely push through the challenges associated with a security driven development track. Understanding that security is a fundamental part of development and injecting that into the cultural of the company will further strength a company's overall health. To do so, there needs to be a clear way for business to understand the security reasoning for various functions of the product and why those functions might be delayed. This will be headed by what was previously discussed in this paper regarding a modeling effort and assets value that is associated with each business process. This will allow the IT and business department to communicate effectively and speak on the same terms.

Future research directions

To future understand the impact of moving toward a security based development cultural there needs to be more research into the cost to switch. There is understanding that the up-front cost will be less but, there needs to be firm numbers for business to act upon. The understanding of this cost will help business shape the entire culture of the company. The research will involve gathering the value of business that have used a model approach to see where they have risk and where they will need to allocate resources to.

Conclusion

With the companies of the world understand that there needs to be more focus on security changing the way we develop software will change the out-come that companies want to have. Understanding that the development of software has to change before companies are held to a new standard will be the next great leap in developing software that is functional and secure at same time.

References used

Jensen, J.; Jaatun, M.G., "Security in Model Driven Development: A Survey," in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on* , vol., no., pp.704-709, 22-26 Aug. 2011

Kasal, K.; Heurix, J.; Neubauer, T., "Model-Driven Development Meets Security: An Evaluation of Current Approaches," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on* , vol., no., pp.1-9, 4-7 Jan. 2011

Lirong Dai; Yan Bai, "An Organization-Driven Approach for Enterprise Security Development and Management," in *Secure Software Integration and Reliability Improvement (SSIRI), 2011 Fifth International Conference on* , vol., no., pp.208-215, 27-29 June 2011

Gregorio, D.D., "How the Business Analyst supports and encourages collaboration on agile projects," in *Systems Conference (SysCon), 2012 IEEE International* , vol., no., pp.1-4, 19-22 March 2012

Hunstad, Amud; Hallberg, Jonas, “Design for secureability – Applying engineering principles to the design of security architectures,” in *Department of System Analysis and IT-Security*, Swedish Defense Research Agency