



# **Survey Results** - Integrating Security into the Software Development LifeCycle

# Survey Results

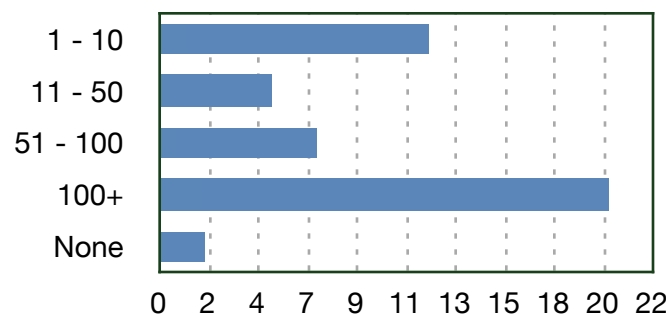
## Summary

Errata Security conducted a survey asking people in the software development community about their experience with integrating security solutions into their Software Development LifeCycle (SDLC). Specifically, the survey focused on adoption of formal software assurance methodologies such as Microsoft SDL, OWASP's SAMM, and BSIMM. The survey was open to the information security and application development communities. This paper will explain the results of that survey. Errata Security welcomes the sharing of the document with anyone interested in application security.

The survey was promoted during the RSA Conference and Security B-Sides San Francisco, as well as on security related blogs. It was open to participation for 2 weeks, and gathered 46 votes. Analysis of these answers can provide a snapshot of the software assurance industry. The survey asked questions about the current practices of software development organizations in terms of adding security related activities to their SDLC methodology. The most value from the survey came from the text box answers where participants were able to describe their experiences. These answers can be summarized by saying that most organizations are looking to increase their security activities. Encouragingly, only 26% of participants answered that they felt including a formal methodology for security would require too many resources. The survey asked the question of whether they were abstaining from these methods for business reason or for lack of awareness. Unlike some of the more technical problems the security industry faces, security SDLC is written in a language familiar to businesses. Therefore, while the implementation of one of these formal security software assurance methodologies was relatively low, 38%, the awareness of one or more of the formal security methodologies was high, 81%.

There was a wide spectrum in the answers, from the size of the organization to the level of awareness of security methodologies. The intention of the survey was to find popular current trends, but it showed that there is no one solution that would be appropriate for every organization. The size of the organization the participant worked at was the most variable. 26% of participants worked on product development teams with less than 10 people, whereas only 43% worked with a team of 100 or more employees. This confirms the assumption that although companies like Microsoft are having the most success creating formal methodologies, they are not representative of the majority of the users.

## Organization Size



There were just as many organizations not using a traditional SDLC as there were organizations using Agile, and there were the same amount of Agile users in 100+ companies as in 1-10 companies. However, there were almost no small companies using Waterfall. A comparison on 100+ companies shows they are evenly split in thirds into Ad Hoc, Agile, and Waterfall. This shows that the software assurance solution for a 100+ company must be different than a 1-10 company, based on their likely methodology.

Also, 86% of the participants answered that their organization sent one or more members of the software development team to security training in a recent or current cycle. However, only 8 out of 46 participants said that their organization sends upper management level employees to training. This is an important distinction because many of these participants explained in other questions that those management employees played a key role in the decision making and the success of the software assurance program. The upper management training factor shows a possible correlation to a more mature security program, since these 8 participants all selected either "Security is always a priority" or "Security is a priority when management dictates." Training was also an interesting factor in the QA Testing department. It was the third highest choice for spending a training budget, over project managers and upper management, in all four organization sizes.

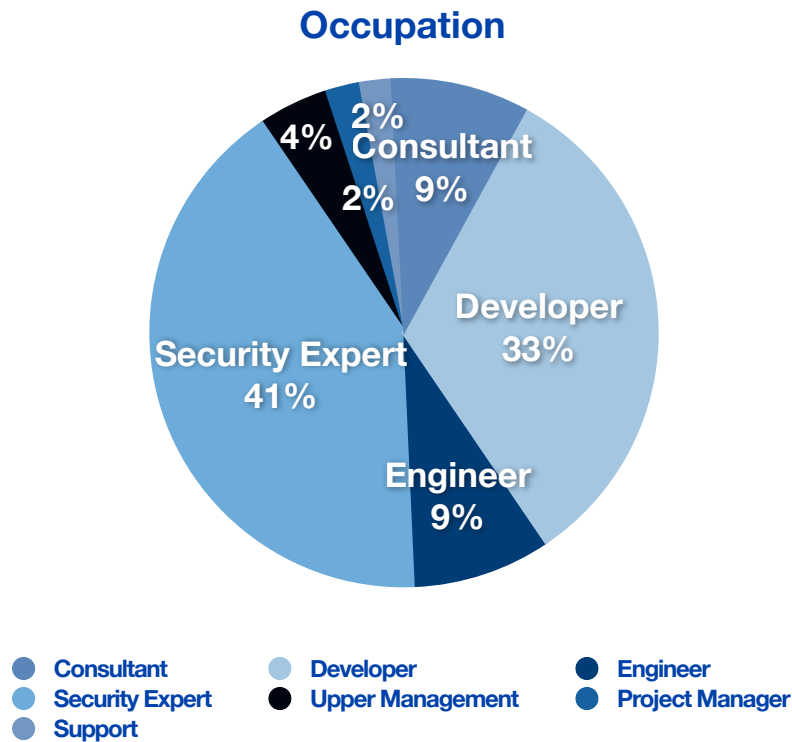
Of particular interest to the security community are the results on testing tools. 16 participants answered that they were using no security tools. At this point in the survey, "security tools" was not defined. It is noteworthy that 6 out of the 7 participants that were able to list their tools by name were using an "ad-hoc" or "custom" security methodology. This shows a high level of tool awareness.

In the follow up questions based on security testing, the survey provided a checklist of all the security activities the organization was using. The most popular was Static Analysis, at 57%, followed closely by Security Code Reviews, at 51%, and Manual Penetration Testing, at 47%. 41% responded yes to "Final Security Review/Audit", and while it was not our intention to illicit information about compliance audits, this question may have given us that information.

## Questions

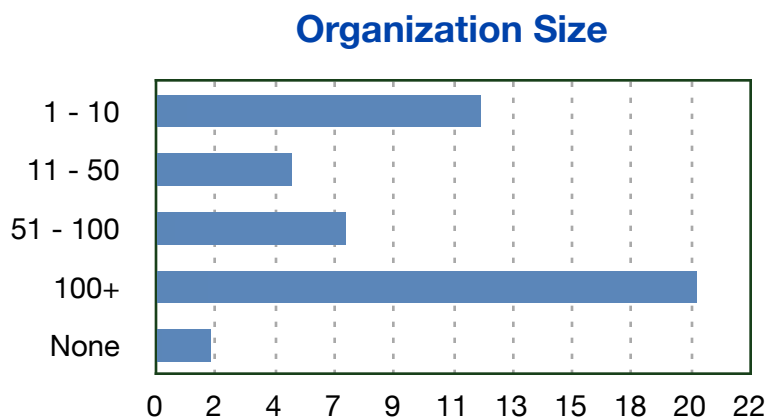
Next, we will go over the specific results of the survey based on each question, and provide analysis of the data.

### Question 1: What job do you provide in the product development process?



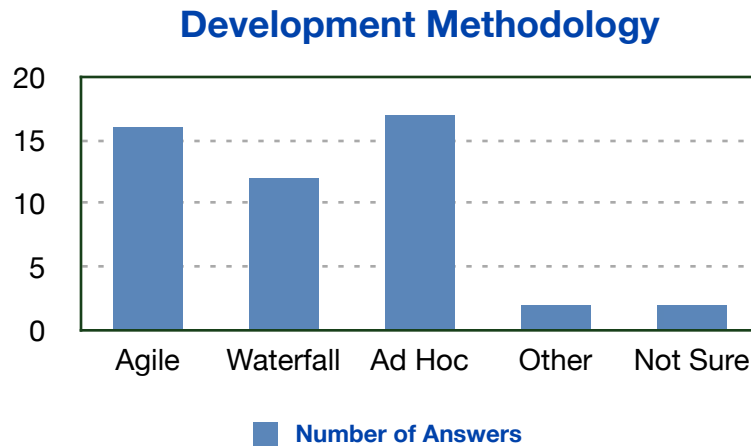
The survey was distributed through a Google Spreadsheet and advertised on blogs, Twitter, and Security B-Sides. Anyone was permitted to take the survey, and there were no participants that were not in the security or software industry. There were 46 responses. Some of these job choices may overlap, so it was up to the participant to self-identify with only one role.

### Question 2: What is the size of your total product development team?



The size of the organization is possibly the most important factor in considering security activities in the SDLC. Larger organizations are able to afford more tools and activities. Although, in question 11, there was still a small amount of 100+ organizations citing a shortage of resources as the reason they abstain from a formal security methodology. While companies such as Microsoft have had great success with their Microsoft SDL, the majority of organizations interested in this field are smaller. Special consideration must be paid to these smaller groups, with a focus on fewer resources requirements.<sup>1</sup>

### Question 3: What development methodology does your department follow?

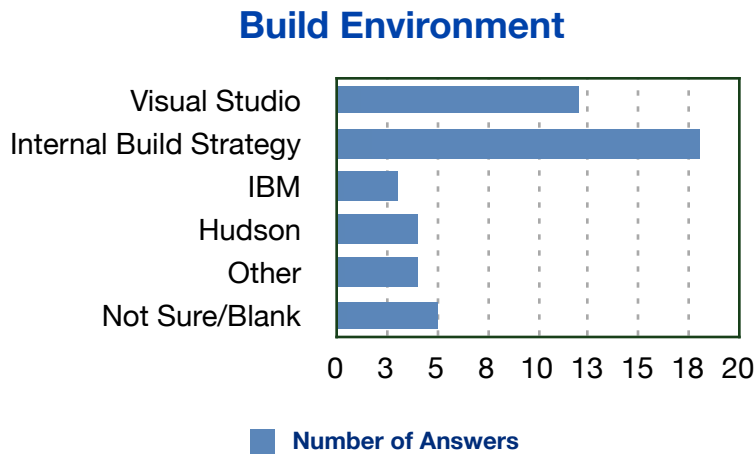


The SDLC development methodology varies greatly from shop to shop. Security advocates have focused most on mapping the Waterfall style to a security framework, but Agile is becoming more supported<sup>2</sup>. This comes from an increase in organizations claiming Agile as their development methodology. Having a security framework that corresponds directly to the development methodology makes adoption easier, but is in no way mandatory. Most organizations are not using a formal SDLC methodology, and in the survey these responses were named “Ad Hoc.” Ad Hoc implementations were spread evenly over all four of the organization sizes. Interestingly, organizations sized 11-50 were the least likely to have an Ad Hoc implementation, and 100+ was the most likely. Methodologies included in “Other” were SCRUM, XP, Spiral, and a combination strategy including two or more of the above.

<sup>1</sup> Resource is broadly defined in this document as “people, time, money, and technology.”

<sup>2</sup> Microsoft announces SDL-Agile for Agile development methodologies - <http://blogs.msdn.com/sdl/archive/2009/11/10/announcing-sdl-for-agile-development-methodologies.aspx>

**Question 4: What build environment does your organization use?**



In order to better understand how to incorporate security into the Development phase of the SDLC, it is important to consider build environments. The Internal Build Strategy was selected by all four organization sizes, but was most common in organizations of 100+ members. The “Other” category included TeamCity and BuildBot.

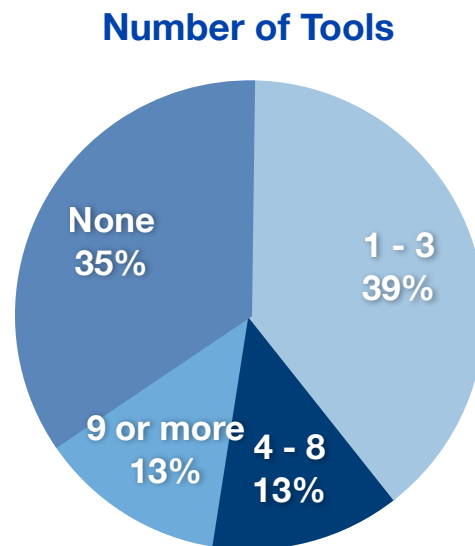
**Question 5: How often is security a concern in the development of your applications?**



In the Security Commitment question, there was no correlation between commitment to security and organization size. Those that said “Only during a security incident” were less likely to do security testing activities than those that answered “Never.” Those that answered “Never” all did some security testing, on average two activities each cycle. 28% selected “Only when it’s a priority from management” implying both that top-down security strategies do have some effectiveness and that people may be looking to management to direct security commitment instead of being self-motivated.

Only half of those that answered “Always” said they were using a formal security SDLC methodology. Whereas we expected to see most of the organizations using a formal security framework to mark “Always,” we found that it was actually only 8 of the 15 organizations. It would seem that without an official metric for the level of commitment, these 7 organizations who did not choose “Always” are underestimating what a large effect this security framework has on software assurance. Or perhaps we can infer here that half of the organizations using a security framework consider it a failure, although this isn’t explicitly stated.

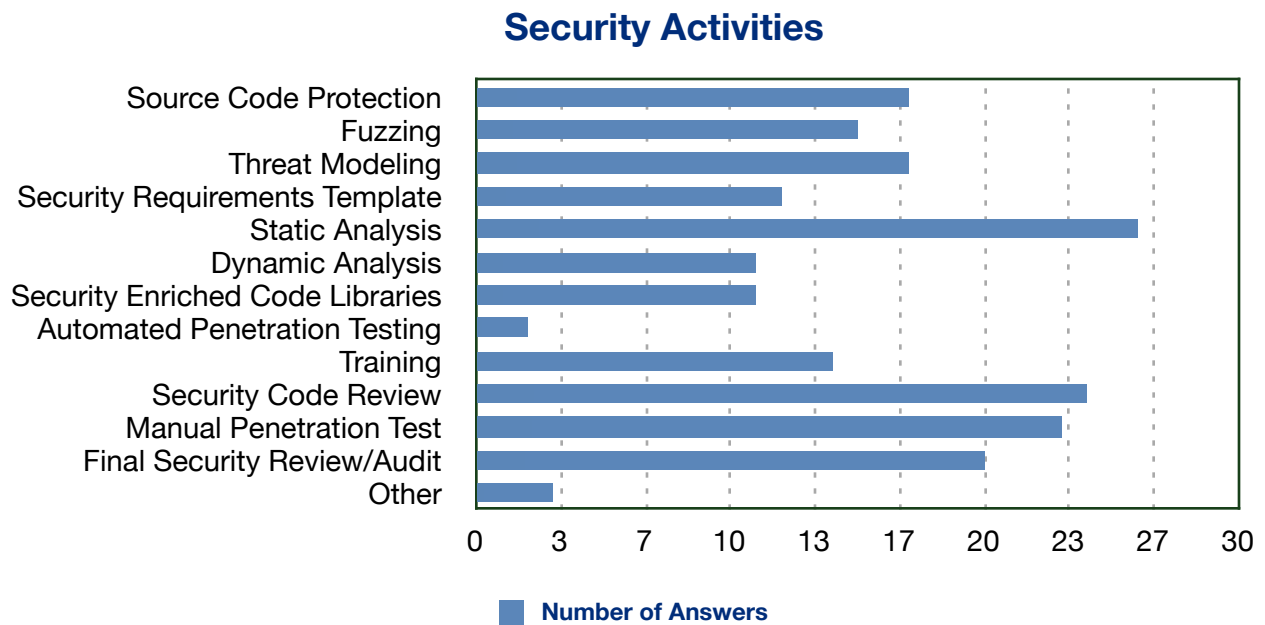
### Question 6: How many types of security tools are used in your organization throughout the software development lifecycle?



The tools listed by participants in the text box associated with this question were AppScan, Burp, HP WebInspect, Hybrid 2.0 (HP AMP + Fortify 360), Tamper Data, SQL Power Injector, XSS Me, NTOSpider, ModSecurity, PHP-IDS, PMD, ratproxy, FindBugs, and custom tools.

There is an anticipated correlation between the number of tools an organization uses and the awareness of formal security methodologies, especially in those that responded “None” to both questions. While security tools are only part of a formal security methodology, they bring awareness across by exposing flaws that must be remediated in later steps in the lifecycle. This connection has already been realized in communities such as the SDL Pro Network, however groups like this are problematic for increasing awareness because a person would have to already be aware of the methodology to find it. Companies making scanners, fuzzers, etc. may be a great source for grassroots awareness.

**Question 7: Please check any box that lists an activity your organization uses in development.**



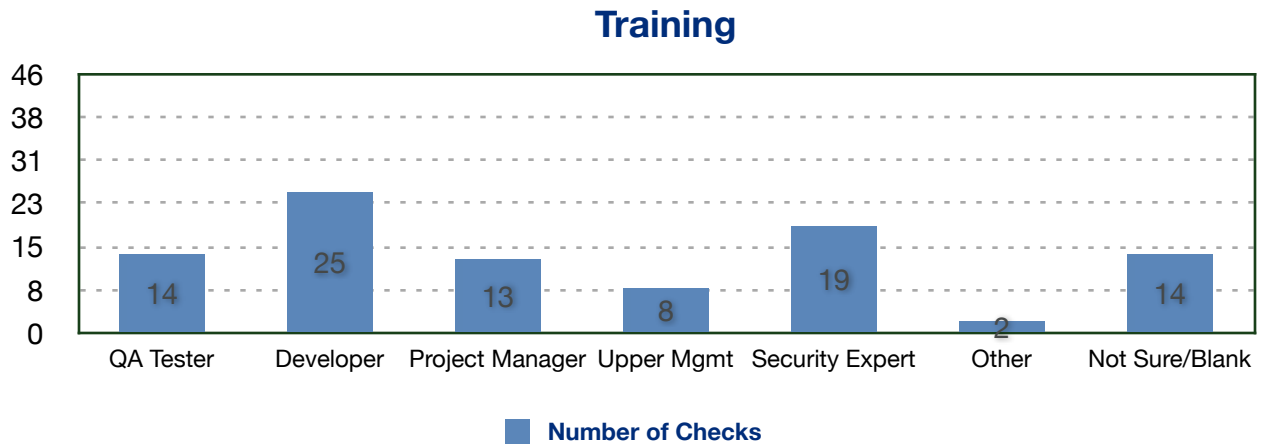
With the exception of one very lucky outlier, small organizations generally don't do many security activities in their application development. Granted not all security activities require equal resources. The popular choices for these organizations were manual penetration testing, security code review, and static analysis.

In the 100+ organizations, there was a variety of answers, but this group had the most activities done on average. Consequently this group also had the most "9+ tools" used in question 6. Static analysis and code review were popular, as with the smaller groups, but this group also added a larger amount of automated activities. It is unclear whether large organizations buy more automated tools because they have a larger budget or because their applications are so complex that it is simply required.

We were also surprised to see how few organizations selected Threat Modeling, at 37%. This activity has shown incredible benefits to increasing security for software assurance, and we expect this number to rise in this year as the method becomes more widely known.

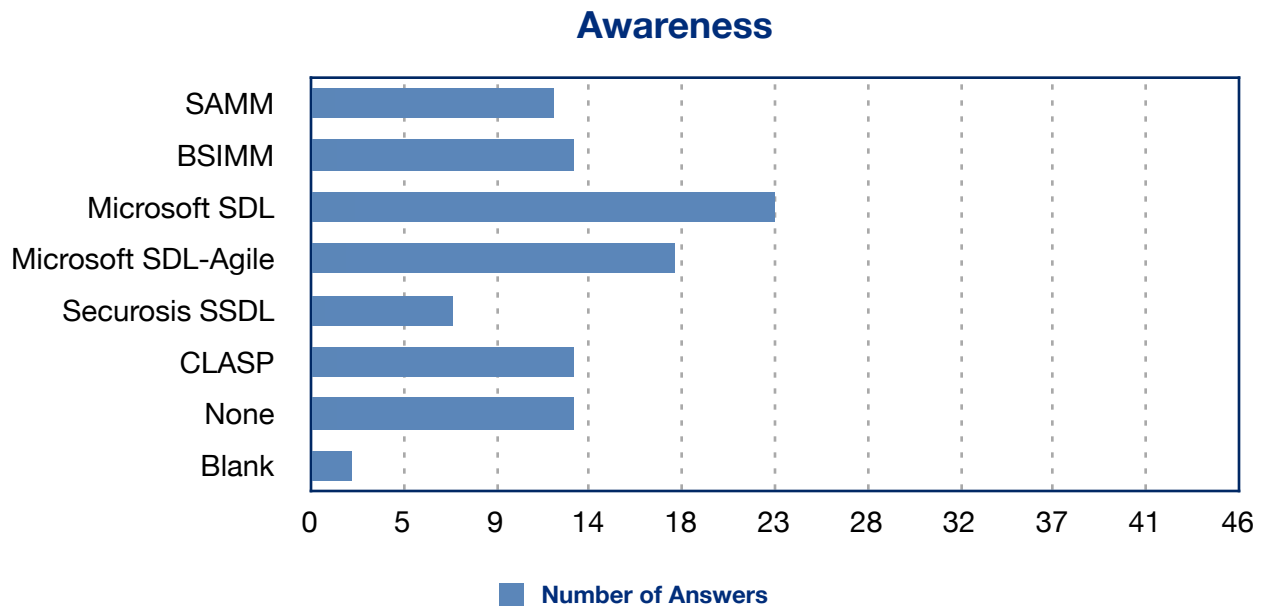
In addition to these activities, participants included compliance, risk management, and web application firewalls.

**Question 8: Please check any box that lists a team member that has received security training during a recent or current development cycle.**



For question 8, we did not define training. We asked only if the training was current. Participants answered that the most of their training hours went to Developers, followed by Security Experts, and then surprisingly QA Testers. At the time of the survey, the role of QA in the security framework was frequently being debated. We expected the impression most participants would have of the training given to their QA team would be much lower. The training for QA Testers was spread fairly evenly across all 4 organization sizes.

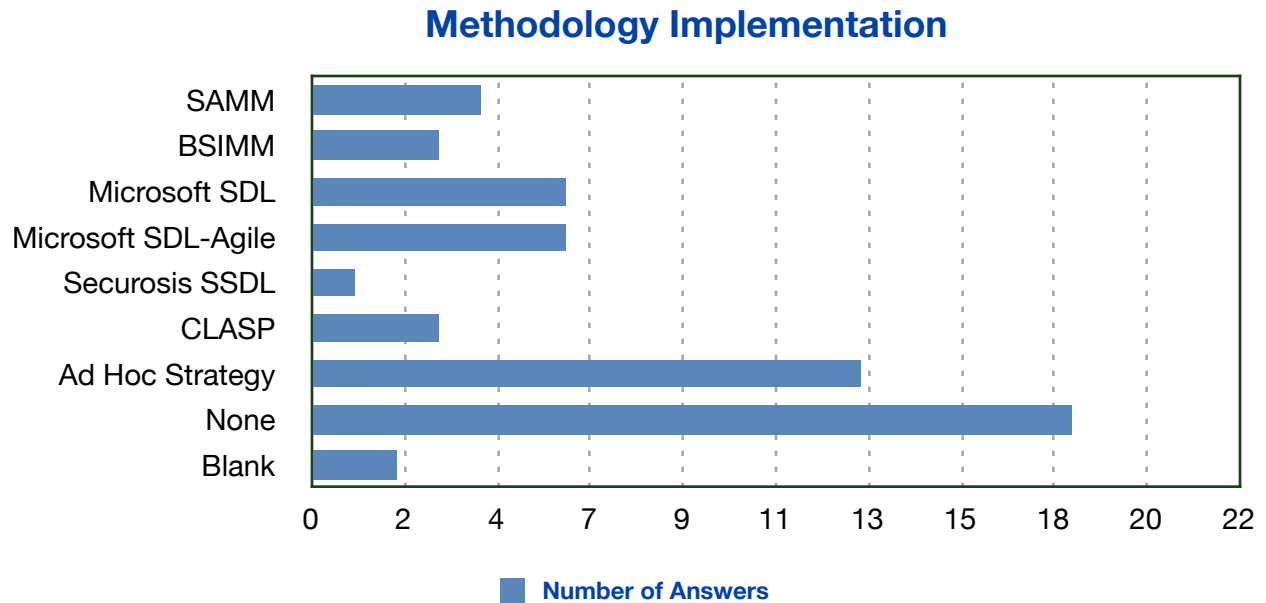
**Question 9: Which security development methodologies are you aware of?**



The intention for this question was to set a baseline for comparison and to see how the software assurance discipline is reaching the application community at large. Out of 46 participants, 23 answered that they were aware of the most well known methodology, Microsoft SDL. Being aware of security tools and activities in application security did not

translate into awareness about security software development lifecycles for half of the participants. In addition to the above, participants added that they were aware of SAFEcode and NIST SP 800-64.

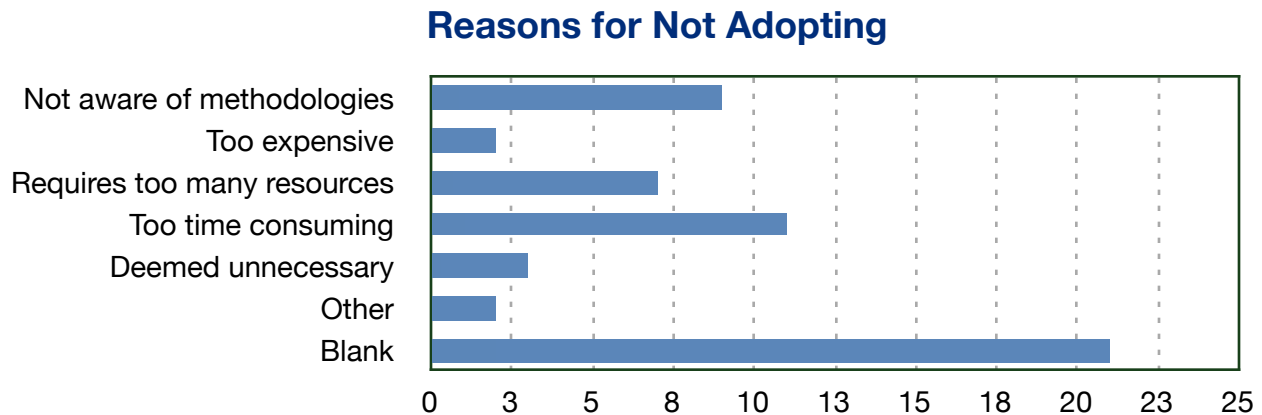
### Question 10: What security development methodology are you currently using in your organization?



There were 15 participants using a formal security framework for the SDLC. The graph shows more answers because many organizations are using a hybrid of methodologies to support a dual Agile/Waterfall environment, for example. Also there seems to be a fine line between the majority Ad Hoc Strategy, and the few that responded with a “custom solution.” This is another example of organizations lacking a bar to compare their implementation against, and thereby either selling their solution short or valuing it too highly. There isn’t enough information in the survey to surmise where this bar would be, i.e. how many security activities would equal a custom framework.

Most of the participants who listed Microsoft Agile considered security to be a priority “Always.” We noticed that many of the Agile users were choosing both Microsoft SDL and Microsoft SDL-Agile, which we attribute to the fact that SDL-Agile was recently released, November 2009, and some organizations are finding the combination easier than migrating.

**Question 11: If you answered None, what was your reason?**



24 participants gave one or several reasons why their organization was not adopting a formal security framework for the SDLC. Other reasons included “low priority” and “too early in startup.”

**Question 12: How are you planning to add security to your software development process?**

METHODOLOGY USED	SIZE	COMMENT
BSIMM, Microsoft SDL, Ad Hoc security tools	100+	We are adding one touchpoint at a time. There is an ebb and flow between adding new touchpoints and enriching existing touchpoints.
Ad Hoc security tools	11 - 50	This is the second time I am working on integrating security within the SDLC from scratch. The first time was from a pen testing/automated testing perspective. This time it was from the foundation up. We have currently deployed Security within the SDLC at a process definition. While we are doing that, we have been doing Secure Code Analysis. The next steps would be to do automated vulnerability analysis.
Ad Hoc security tools, BITS FISAP, NSA IEM, DIACAP, NIST SP 800-64, Parts of Open SAMM and the Microsoft SDL (SDL-Agile soon)	100+	That's my job. I take it very seriously. Most of the time, it's working in "the shit" with developers. I have no idea how many times I've explained parameterized/callable queries with proper variable binding -- perhaps about as many rows as there are in the database in question.
Ad Hoc security tools	100+	Planning to move security further "left" in the cycle. Unfortunately my executive management is more concerned with getting a product out the door than getting a secure product out the door. Until that changes, I don't know how successful I can be...
None	1 - 10	Need to get more information about tools, methodologies, etc. We are a very small team. All this is very time and money consuming (my feeling). Need to find out the best approach (money/time)
None	1 - 10	Most of the development is R&D which gets into production. So security is mostly from the developer's observation that the implementation might cause some security issue. Other than that until unless some break in happens, security in SDLC is least concern
None	100+	Contact with a specialist.

METHODOLOGY USED	SIZE	COMMENT
None	1 - 10	<i>Constant conversation and review of our practices as we try to figure out what we can do while dealing with extremely limited resources and no budget.</i>
Ad Hoc security tools	1 - 10	<p><i>As we focus on web and REST environment applications our security focus is generally limited to:</i></p> <p><i>Identifying and mitigating possible XSS holes</i>  <i>Input validation and cleaning to avoid SQL injection</i>  <i>Data validation and client data protection</i></p> <p><i>As our servers are not hosted in house the physical, network and core system security is supposed to be handled by our provider. How well they actually do that is still open to question as (of course) I am not in a position to pen-test anything other than our own applications/servers, and then only after clearing with the hosting company so they know who is attacking and what the target is. (A bit like the health dept calling a restaurant to say there will be a "surprise inspection" next Tuesday at 3PM.)</i></p>
SAMM, Microsoft SDL-Agile	11 - 50	<i>Additional focus on threat modeling and fuzzing.</i>
Ad Hoc security tools	100+	<i>As part of the QA.</i>
None	100+	<ol style="list-style-type: none"> <li><i>1. Added web app vulnerability test results to executive performance scorecard</i></li> <li><i>2. Added security training for app dev teams to executive performance scorecard</i></li> <li><i>3. Publicizing security methodologies, tools and support organizations (OWASP, etc) to the app dev teams</i></li> <li><i>4. Delivering instructor-led and webinar sessions to app dev teams</i></li> </ol>
The Principles of Secure Development	50 - 100	<i>We are always trying add more security into our SDLC. We have it integrated into each phase but we are now attempting to supplement the manual process (i.e. security code review) with automated tools such as Fortify and HP WebInspect</i>
None	1 - 10	<p><i>As a former security engineer/consultant, security is present throughout the development process as a side-effect. We are a small shop developing web applications. In some cases, frameworks (such as Rails) that address common security concerns are used. In other cases, when custom PHP frameworks are in use, I perform ongoing checks with my co-founder to ensure that he is aware of potential risks and has addressed them properly.</i></p> <p><i>Once development is complete, I perform both a code review and QA/security testing to ensure that we have addressed anything as necessary.</i></p>
SAMM, Microsoft SDL	100+	<i>Already built in</i>

This is a selection of quotes from the survey. The friction between the business priorities of management and the technical priorities of developers and security experts is evident. It is exciting to see that a large company has decided to increase their security in application development by increasing the QA phase. This is a good indicator for the overall posture of security in application development currently.

## Conclusion

While it seems there is a connection between organization size and methodology adoption, the numbers do not strictly agree. There is a barrier to entry for small organizations, but the medium to large organizations' adoption varies greatly. It's true that current formal methodologies do require a resource investment, but if that was the only factor then we would see a steadily increasing line as the company size increased. Other factors, such as lack of awareness and lack of management buy-in also contribute to organizations abstaining from the process. In order to increase adoption, we need to customize the procedure and create awareness around affordable implementations.

So where do we go from here? If you're a small company, the good news is that any little bit does count. A startup does not have to give up the ability to generate secure code in their first development lifecycle and miss out on all the cost saving benefits down the road. More than half of these small companies are using Agile development, and recently the software assurance community has generated a lot more support and content for an Agile security development methodology. In this community, there are affordable tools<sup>3</sup> and trainings available.<sup>4</sup> The survey showed that companies with 1-10 employees were willing to train outside of the developer room. By spreading the training to the QA Testers, they were able to create a more valuable asset to the company, and increase security by defense in depth. The small size of the organization should be an advantage to promote the chance of success for the security program. Initiatives like Rugged Software<sup>5</sup> are free ways to attack the problem of software vulnerabilities at the people level. With the ability to get creative about the solution, small organizations can be flexible enough to increase security without straining resources.

If you're a large software development company, the decision matrix is more complex, and the stakes are much higher. Compliance may be an issue. The SDLC methodology market is saturated with dozens of choices, all with varying levels of support. Organizations of this size are usually running at least two different security development methodologies together to find the correct set of activities. Customization is critical. Studies like BSIMM show that an a la carte style of security activities fitting the organization are successful for 30 large companies. In Errata's survey, that style qualifies as the "Ad Hoc Strategy." However, companies like Microsoft and OWASP have shown that Six Sigma-style benefits can be measured if an organization goes higher in the maturity model. The earlier you incorporate security practices into your product development, the more resources you save over all. Threat Modeling is an essential step to gain those benefits. The number of organizations doing Threat Modeling is 37% according to the survey, but we expect this number to increase dramatically as it has recently received a burst of support in the community. Another strategy to achieve the benefits is bringing the QA team into earlier phases. As described for small organizations, the same goes doubly for large ones. If the QA team is trained in security testing, they provide accountability for vulnerabilities that they discover. It shortens the process for the developer team, and in the Verification phase they will now have the more dynamic role of verifying the security bugs have been fixed.

In the future, we can expect to see many more security development methodologies pop up in the same way that the SDLCs did before. This will allow organizations to find the methodology that is the right fit for their needs. As these ideas disseminate outside of the security community, more security professionals from all areas of information security will be expected to know about these methodologies. Due to compliance issues and customer mandates, this is a field of security that will grow quickly in the coming years.

---

<sup>3</sup> Errata Security offers a suite of testing tools for free at <http://www.erratasec.com>

<sup>4</sup> For example, Microsoft has the SDL Pro Network at <http://www.microsoft.com/security/sdl/getstarted/pronetwork.aspx>

<sup>5</sup> Rugged Software Development Initiative is a value system encouraging secure coders <http://www.ruggedsoftware.org>