How to Add Security Requirements into Different Development Processes
# Table of Contents

1. **Introduction** ........................................................................................................................................... 3
2. **Current State Assessment** ......................................................................................................................... 4
3. **Waterfall-Style Development** ..................................................................................................................... 5
   3.1. Description ........................................................................................................................................... 5
   3.2. Waterfall Roles & Responsibilities ........................................................................................................ 6
4. **Agile-Style Development** .......................................................................................................................... 7
   4.1. Description ........................................................................................................................................... 7
   4.2. Agile Roles & Responsibilities ................................................................................................................ 8
5. **Continuous Development / No Process** ...................................................................................................... 9
   5.1. Description ........................................................................................................................................... 9
   5.2. Continuous Development Roles & Responsibilities ................................................................................ 10
6. **Conclusion** .................................................................................................................................................... 11
1. Introduction

Embedding software security requirements into development teams can be complicated. Organizations often struggle to understand how to change their development processes to include requirements.

Very few organizations with multiple development teams have a single, standard development process. Instead, development teams tend to build software the way that works best for them and their business needs. We have observed that while each team's development process may differ in details, most teams tend to fall into one of three patterns of development:

- Agile
- Waterfall
- Continuous development

This whitepaper outlines suggestions for embedding security into each, allowing individual development teams to select the method that works best for them. In order to use these processes you must first have a program in place to generate software security requirements. Read our whitepapers: 5 Steps to Starting a Software Security Requirements Program and Automated Scaling of Security Requirements.

As with any process guidance, you will need to change elements of this document to incorporate your organization's unique culture, nomenclature, and processes prior to using the guidance.
2. Current State Assessment

We recommend every existing application should undergo a current state assessment of security requirements as a first step, prior to changing the development process. A current state assessment involves the following steps:

• Define security requirements for an application. For example, by modeling the application in SD Elements.

• Review resultant list of requirements over a certain risk threshold (e.g. all high risk/priority requirements). Identify all requirements that have not been implemented or that the development team is not sure have been implemented. This is the set of in-scope requirements. Explicitly identify requirements where the business is willing to accept the risk not completing these requirements.

• Add in-scope requirements to the appropriate requirements repository. Generally this is a defect tracking or application lifecycle management (ALM) system. In less structured teams, this may simply mean e-mailing a lead developer with the set of security requirements or adding them to a spreadsheet.

Remember that each requirement adds overhead to the development process. We recommend limiting scope to high risk/priority requirements to begin with.
3. Waterfall-Style Development

3.1. Description

This development style breaks software development into distinct phases and generally has different stakeholders involved in each phase. Waterfall development styles generally include up-front planning phases such as requirements gathering. Contrasted with agile development, waterfall development tends to have large release cycles (e.g. once every quarter, bi-annually, or annually). Note that some organizations use a hybrid agile-waterfall approach where they develop software in short sprints/iterations but still have large software releases. For the purposes of the embedding security requirements, such agile-waterfall hybrid teams should follow the guidance in the waterfall-style development model.

The following diagram depicts a typical, simplified waterfall development process with security requirements integration.

Process Diagram

We recommend embedding security requirements roughly at the process steps listed above. After the Current State Assessment, you should periodically review application security requirements according to their risk profile. We recommend the following frequencies:

<table>
<thead>
<tr>
<th>Application risk profile</th>
<th>Frequency of review</th>
</tr>
</thead>
<tbody>
<tr>
<td>High risk applications. For example, external facing web applications with credit card data</td>
<td>Every major release, or every quarter (whichever is more frequent)</td>
</tr>
</tbody>
</table>
Medium risk applications. For example, internal applications storing / processing Personally Identifiable Information (PII) | Twice per year
---|---
All other applications that are subject to information security review | Once per year

### 3.2. Waterfall Roles & Responsibilities

The following table outlines the stakeholders involved in using security requirements within a development process, and their suggested responsibilities.

<table>
<thead>
<tr>
<th>Role</th>
<th>Initiate process</th>
<th>Review requirements</th>
<th>Decide on requirements in-scope for release</th>
<th>Assign requirements to developers</th>
<th>Perform testing tasks</th>
<th>Follow-up on requirements progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information security analyst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Development lead</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project manager</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Assurance (QA) lead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
4. Agile-Style Development

4.1. Description

This development style uses short iterations to build software incrementally. While there are several specific types of agile development, such as Scrum and Kanban, they share certain characteristics:

- Iterations are short, generally from 1 to 4 weeks in length
- Iterations begin with short planning meetings
- Every iteration involves building shippable software
- Practitioners tend to push back on unnecessary process overhead

The following diagram depicts a typical agile development process with security requirements process integration.

**Process Diagram**

We recommend embedding security requirements at the process steps listed above. Add security requirements to the product backlog and allow development teams to select when the requirements should be in-scope. After the Current State Assessment, you should periodically review applications in security requirements according to the risk profile. Add any newly-introduced requirements into the product backlog. We recommend the following frequencies:

<table>
<thead>
<tr>
<th>Application risk profile</th>
<th>Frequency of review</th>
</tr>
</thead>
<tbody>
<tr>
<td>High risk applications. For example, external facing web applications with credit card data</td>
<td>Every sprint / iteration</td>
</tr>
</tbody>
</table>
Medium risk applications. For example, internal applications storing / processing Personally Identifiable Information (PII) | Twice per year
---|---
All other applications that are subject to information security review | Once per year

### 4.2. Agile Roles & Responsibilities

The following table outlines the stakeholders involved in using security requirements within an agile development process, and their suggested responsibilities.

<table>
<thead>
<tr>
<th>Role</th>
<th>Initiate process</th>
<th>Review requirements</th>
<th>Decide on requirements in-scope for backlog</th>
<th>Assign requirements to iteration</th>
<th>Perform testing tasks</th>
<th>Follow-up on requirements progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information security analyst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Product owner</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrum master / project manager</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Assurance (QA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
5. Continuous Development / No Process

5.1. Description

This development style uses short iterations to build software incrementally. While there are several specific types of agile development, such as Scrum and Kanban, they share certain characteristics:

Many modern fast-moving development teams have removed the concept of release planning altogether. Instead, the organization maintains a list of features & defects and developers build, test, and deploy continuously using automated deployment processes. This means that security teams cannot review release / iteration development goals to embed security. On the other hand, the time to deploy a fix or change is much lower than other development styles. In other cases, development teams do not follow any development process at all. In these teams, requirements are not formally tracked in any system.

In both of these cases, the best option is to regularly review Security requirements retroactively against recent changes while continuously embedding security controls into the application's framework.

The following diagram depicts a typical continuous development process with Security requirements process integration.

Process Diagram

With this development style, you must periodically review application security requirements according to the risk profile. We recommend the following frequencies:

<table>
<thead>
<tr>
<th>Application risk profile</th>
<th>Frequency of review</th>
</tr>
</thead>
<tbody>
<tr>
<td>High risk applications. For example, external facing web applications with credit card data</td>
<td>Bi-weekly</td>
</tr>
</tbody>
</table>
Medium risk applications. For example, internal applications storing / processing Personally Identifiable Information (PII)  

<table>
<thead>
<tr>
<th>All other applications that are subject to information security review</th>
<th>Once per year</th>
</tr>
</thead>
</table>

### 5.2. Continuous Development Roles & Responsibilities

The following table outlines the stakeholders involved in using security requirements within a development process, and their responsibilities.

<table>
<thead>
<tr>
<th>Role</th>
<th>Initiate process</th>
<th>Review requirements</th>
<th>Decide on requirements in scope for release</th>
<th>Assign requirements to developers</th>
<th>Perform testing tasks</th>
<th>Follow-up on requirements progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information security analyst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Development lead</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Quality Assurance (QA) lead</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
6. Conclusion

Successfully deploying security requirements can yield dramatic results: applications built faster with fewer security vulnerabilities, improved communication between development and security teams, and increased assurance that you are aware of known security risks you are taking when deploying code.

Like any other process change, embedding software security requirements is not easy. Development teams are optimized to deliver value to their business, and asking them to alter their process for the sake of security is often a losing battle. By following the practices outlined in this paper, you can dramatically reduce the burden of development teams to integrate software security. At the same time, you can save thousands of dollars in remediation and significantly reduce the risk of costly breaches.

While the steps above discuss generalized process integration, remember that technical details matter as well. In particular, how you present security requirements to developers can dramatically alter the chance of them being adopted. Our experience is that adding security into application lifecycle management (ALM) tools is the most effective method of adoption.
About SD Elements

SD Elements is your guide for secure software development. Be more proactive with automated requirements generation that scales quickly. Make security measurable with clear links between requirements & test. Proactively eliminate up to 97% of application security risks by building more secure software from the start.

Learn more at www.sdelements.com