A situation analysis of the security awareness at Software Vendors and how to best inform them about the Microsoft Security Development Lifecycle

av

Johannes Gunnbäck
Helena Mischel

LIU-IDA/LITH-EX-A--09/065--SE

2010-01-21
Examensarbete

A situation analysis of the security awareness at Software Vendors and how to best inform them about the Microsoft Security Development Lifecycle

av

Johannes Gunnbäck
Helena Mischel

LIU-IDA/LITH-EX-A--09/065--SE

2010-01-21

Handledare: Kristian Sandahl
            IDA, Linköpings universitet
            Johan Lindfors
            Microsoft

Examinator: Kristian Sandahl
            IDA, Linköpings universitet
A situation analysis of the security awareness at Software Vendors and how to best inform them about the Microsoft Security Development Lifecycle

In January 2002 Bill Gates sent out the renowned ‘Trustworthy Computing’ memo where he announced that the company would shift their focus from adding new features and functionality to security and privacy. This was what led to the formulation of the Security Development Lifecycle (SDL). This process is now mandatory for all development at Microsoft with meaningful business risk and/or with access to sensitive data. The SDL led to great improvements of the number and severity of vulnerabilities in the products that went through the process. When the vulnerabilities in the Operation System (OS) were diminished Microsoft noticed that the threats moved to the application layer. This led to them wanting to spread their model to application developers. One interesting target group is mid-sized Independent Software Vendors (ISVs), mainly because there are so many of them. Finding out what development process they use today and how they would benefit from and could be informed about the SDL is of interest for Microsoft.

Interviews with Microsoft evangelists, security experts and representatives from the target group has been preformed to get a better understanding of the situation today and how it could be improved. The interviews have resulted in a number of recommendations for how to adjust the SDL and the information concerning the process to meet mid-sized ISVs needs. A clear need for information, that is categorized and directed to the different business areas in the software industry, with specific recommendations and courses of action for each of them, has been identified. The interviews have also resulted in a situation analysis of the security awareness at the target group today and the experts view of what activities in the SDL they would benefit from. The maturity level amongst the ISVs was found to be low and their own estimated vulnerability level was low. The estimated security awareness in the future on the other hand is high, this can be accounted for the upcoming migration to cloud services that is requested by the customers and the security issues this will lead to. One thing that is agreed upon that would be suitable to introduce is threat modeling. This requires little security knowledge yet leads to dramatic reduction in vulnerabilities. The experts have also shared improvements they think could be made on the SDL.

Nyckelord

Keywords SDL, Security, Development, ISV
Abstract

In January 2002 Bill Gates sent out the renowned "Trustworthy Computing" memo where he announced that the company would shift their focus from adding new features and functionality to security and privacy. This was what led to the formulation of the Security Development Lifecycle (SDL). This process is now mandatory for all development at Microsoft with meaningful business risk and/or with access to sensitive data. The SDL led to great improvements of the number and severity of vulnerabilities in the products that went through the process. When the vulnerabilities in the Operation System (OS) were diminished Microsoft noticed that the threats moved to the application layer. This led to them wanting to spread their model to application developers. One interesting target group is mid-sized Independent Software Vendors (ISVs), mainly because there are so many of them. Finding out what development process they use today and how they would benefit from and could be informed about the SDL is of interest for Microsoft.

Interviews with Microsoft evangelists, security experts and representatives from the target group has been performed to get a better understanding of the situation today and how it could be improved. The interviews have resulted in a number of recommendations for how to adjust the SDL and the information concerning the process to meet mid-sized ISVs needs. A clear need for information, that is categorized and directed to the different business areas in the software industry, with specific recommendations and courses of action for each of them, has been identified. The interviews have also resulted in a situation analysis of the security awareness at the target group today and the experts view of what activities in the SDL they would benefit from. The maturity level amongst the ISVs was found to be low and their own estimated vulnerability level was low. The estimated security awareness in the future on the other hand is high, this can be accounted for the upcoming migration to cloud services that is requested by the customers and the security issues this will lead to. One thing that is agreed upon that would be suitable to introduce is threat modeling. This requires little security knowledge yet leads to dramatic reduction in vulnerabilities. The experts have also shared improvements they think could be made on the SDL.
Acknowledgments

This section is dedicated to everyone that has been of help to us in the work with this thesis. We especially want to thank the security experts John Wilander, Sergio Molero and Stuart Glasson who helped us in getting a better understanding on how security in development works today and gave us a lot of useful ideas and tips. We also want to thank the following people:

Johan Lindfors, our supervisor at Microsoft who guided and helped us and the other co-workers at Microsoft that took part of our survey. Kristian Sandhal, our supervisor at the university. The interviewees at the companies of our target group, that took the time to answer our questions, Cybercom, IFS, Medius, Prevas. David Taylor and Erik Sigvardsson, our opponents, who reviewed the report. Finally we want to thank our family and friends for their support during the thesis and during the years of our education at Linköpings Tekniska Högskola.

Johannes Gunnbäck
Helena Mischel
# Contents

1 Introduction .......................... 1
   1.1 Background ............................. 2
   1.2 Purpose ................................. 2
   1.3 Questions at issue ..................... 2
   1.4 Boundaries .............................. 2
   1.5 Outline .................................. 3
   1.6 Bias in information ..................... 3

2 Method .................................. 5
   2.1 Literature Study ......................... 6
   2.2 Deep Interviews ......................... 6
   2.3 Questionnaire ............................ 6
   2.4 Method criticism ......................... 7

3 Software Security ........................ 9

4 The Security Development Lifecycle .......... 13
   4.1 SDL History ............................... 14
   4.2 The SDL Process .......................... 14
      4.2.1 Training ............................... 15
      4.2.2 Requirements ........................... 17
      4.2.3 Design ................................ 19
      4.2.4 Implementation ......................... 21
      4.2.5 Verification ............................. 22
      4.2.6 Release ................................. 23
      4.2.7 Response ................................ 24
   4.3 SDL-Agile ................................ 24
   4.4 SDL and Management ....................... 27
   4.5 Does the SDL work? ....................... 28
   4.6 Spreading the word ....................... 32
   4.7 SDL and the Software Business ............ 34

5 Security Development in the Software Business .... 37
   5.1 The View on Security at Microsoft ........ 38
| 5.1.1 | Development at the ISVs | 38 |
| 5.1.2 | Information Flow and Influence | 39 |
| 5.1.3 | The Future | 39 |
| 5.1.4 | The SDL | 40 |
| 5.2 | The View on Security from the Experts | 40 |
| 5.2.1 | Development at the ISVs | 41 |
| 5.2.2 | Information Flow and Influence | 41 |
| 5.2.3 | The Future | 43 |
| 5.2.4 | The SDL | 43 |
| 5.3 | The View on Security at the ISVs | 44 |
| 5.3.1 | Development at the ISVs | 44 |
| 5.3.2 | Information Flow and Influence | 45 |
| 5.3.3 | The Future | 46 |
| 5.3.4 | The SDL | 46 |

### 6 Analysis & Discussion

6.1 Why Security in the Development Processes? | 50
6.2 Situation Analysis | 50
6.3 The Relevance of the Microsoft SDL | 51
6.4 Where and How can Changes be Made? | 52

### 7 Results

7.1 How is the term security defined in the context of secure development processes? | 56
7.2 How does the SDL make products more secure? | 56
7.3 What is the security experts view of secure development amongst mid-sized ISVs today? | 57
7.4 What changes would mid-sized ISVs benefit from? | 57
7.5 Further work | 58

### Bibliography

59

### A SDL-Agile Requirements

63

### B Microsoft SDL: Return on Investment, Project Example

68

### C Interview Questions

72
Chapter 1

Introduction

This is a master thesis made at The Institute of Technology at Linköping University for The Department of Computer and Information Science. The thesis is a part of the Master of Information Technology programme. This thesis concerns security in development processes and is performed on the behalf of Microsoft.
1.1 Background

Many of the larger software development companies had to find out about the importance of security the hard way, ie after detection of vulnerabilities or attacks, to realize how important it is to get security in the development lifecycle. Having a Security Development Life Cycle (SDLC)\(^1\) reduces vulnerabilities early in the development process and aims to having as few flaws as possible at release. Microsoft started "The Trustworthy Computing Initiative" in early 2000, which led to the Security Development Lifecycle (SDL). The process is now used in all development, with meaningful business risk and/or with access to sensitive data, at the company. Today, attacks on Operation Systems (OSs) have declined, and instead have moved to the application layer that indirectly affects the Microsoft platform. Microsoft would like to share their knowledge and experiences on security development. They hope that application manufacturers will adopt the SDL and thus get the same drastic reduction of vulnerabilities they have experienced since the introduction of the SDL. A lot of applications are developed by smaller companies and reaching out to them is therefore of particular interest.

1.2 Purpose

The purpose of this thesis is to perform a situation analysis of the security awareness at mid-sized Independent Software Vendors (ISV). Based on this, find a way to fit the SDL into the picture and propose what actions are needed to cause them to adopt the process.

1.3 Questions at issue

- How is the term security defined in the context of secure development processes?
- How does the SDL make products more secure?
- What is the security experts’ view of security development amongst mid-sized ISVs today?
- What changes would mid-sized ISVs benefit from?

1.4 Boundaries

Our target group in this thesis is mid-sized ISVs. We chose this group together with Microsoft since it was the most interesting group and it would be a good

\(^1\)SDLC is the general term for a security process.
indicator on the rest of the software industry’s state. The reason for this is that they are many of them and they play a big role in the industry. Thier multitude and the fact that Microsoft has a good contact with them also helped with getting the adequate data in the empirical study of this thesis. The objective of this thesis is not to educate the reader in how existing development processes and SDLCs are performed in detail but to highlight what’s used, and why it’s used. Security in development processes is a rather new topic and changes are made frequently. Over the time we have worked with this thesis new information has appeared and therefore we can’t guarantee that all the details are up to date.

1.5 Outline

The outline of this thesis is divided into six sections; introduction, method, theory, empirical study, analysis and discussion, and result. In the introduction part we describe the background and purpose for this thesis together with boundaries and source criticism. We continue with the method chapter where we describe how we conducted our study. Then we present the theory, divided into two chapters; one which explains the term security within the context of security development, and one that describes the Microsoft SDL and its impact. The theory chapter is followed by the empirical study. Here we show the results from our situation analysis and these are analyzed and discussed in the "Analysis & Discussion" chapter. We round off with our conclusions in the result chapter.

1.6 Bias in information

The first thing that will come to mind when you read this is that there is a lot of "Microsoft this..." and "Microsoft that..." and on top of that there are also a lot of facts referring to Microsoft sources which can sound a bit partial. The problem has been there isn’t that much information about the SDL outside of Microsoft. Some information has also been taken from sources like blogs or presentation materials which may not be as reliable as books, white papers and scientific publications. Additional criticism is that, although they were chosen at random, there is always a risk that the interviewees don’t represent our target group well enough. The reason could be that the company culture or their own interest level in the matter differs from most of the other mid-sized ISVs. Our initial perspective on security development may also have influenced the angle of this thesis.
Chapter 2

Method

In this chapter we will go through the methods we chose to elicit our empirical data and the reason why we have chosen them. We also present some criticism and alternative methods to the ones we have selected.
2.1 Literature Study

The literature study is mainly based on books, white papers and online articles. These are complemented with information from blogs, podcasts and presentation material. We started off with reading a vast amount of material to get a solid base of knowledge on the subject. The material was found through the SDL homepage, tips from security lecturers and security experts, and from simply google searches on "SDL". We ended up with a lot of sources which we sifted through for the theory part of the thesis. We have tried hard to be diverse in the usage of sources so that the theory will have veracity.

2.2 Deep Interviews

The elicitation technique we used is semi-structured interviews. These were chosen after reading about interviewing methods in *Intervju som metod* [2] and consulting with our supervisor at Linköping University. This helped us conclude that they were the most suitable for our objective because we could get as much information as possible from each interviewee. All of the interview sessions were at least one hour long and included 40-60 questions (Appendix C). We began our interview series with five loosely structured sessions with some key persons at MS. These work with our target group on a day to day basis. From these interviews we formed the base material for our further interviews. The next step in the process was to interview security experts. The three security experts we interviewed also have a lot of contact with our target group because they work as security consultants at these companies and teach them about security. We ended the series of interviews with four employees on four different companies from our target group, mid-sized ISVs, in most cases the security manager. The interviews with the experts and the ISVs were validated through mail communication where they got a copy of the summary of their interview and where asked to respond if they had any complaints. We didn’t have to change anything after the validation. The ones with the key persons we didn’t feel the need to validate; they were mostly carried out for our own sake. We instead conciliated the results with our supervisor at MS.

2.3 Questionnaire

After the interviews we wanted to supplement the information we got from the three security experts and make it more distinct. So we sent them a questionnaire specifying all the activities in the SDL and asking them to grade the relevance for our target group from high to low. We got replies from two out of three. This was also meant as an extra guideline for the ISVs.
2.4 Method criticism

We could have interviewed even more people but were limited by the time frame of the thesis. We could also have made a quantitative survey to complement our qualitative material but unfortunately we did not get access to the right forum to perform such an evaluation. To get a more free and open discussion we chose, after consulting our supervisor at Linköpings University, to not tape our interviews. Instead we took notes which we later validated. For more scientific rigour we could have used some kind of recording instrument and transcribed the recordings to open for independent replication of our analysis.
Chapter 3

Software Security

The following question arised frequently during our work with this thesis "What do you mean by security?". This lead us to put aside this section to clarify what security means for us and this thesis.

"By security I don't mean security features, I mean secure features."
-Michael Howard
Security has for long referred to features like firewalls, logins, ssl, certificates, cryptology etc. When we talk about security in this thesis we mean security in the development process. With this we refer to security that comes "from the ground up" and involves the whole development lifecycle and not just security features as mentioned above. Instead it involves things like security education, non-functional security requirements, design principles, security and privacy testing and having a response plan when something goes wrong.

Dieter Gollman says, in his book Computer Security [4], that security is about protecting your assets. To do this protective measures are needed which he classifies as:

- **Prevention**: take measures that prevent your assets from being damaged.
- **Detection**: take measures that allow you to detect when an asset has been damaged, how it has been damaged, and who has caused the damage.
- **Reaction**: take measures that allow you to recover your assets or to recover from a damage to your assets.

To capture the notion of computer security, Gollman lists aspects of how information assets can be compromised, which are called the CIA-criterias:

- **Confidentiality**: unauthorised disclosure of information.
- **Integrity**: unauthorised modification of information.
- **Availability**: unauthorised withholding of information or resources.

The CIA-criterias are also what Microsoft base their definition of a secure product on: "a product that protects the confidentiality, integrity, and availability of the customers’ information, and the integrity and availability of processing resources, under control of the system’s owner or administrator" [17]. Confidentiality can be divided into two areas, privacy and secrecy. Privacy is about protection of personal data and secrecy is about protection of data belonging to an organization. [4]

Privacy is a huge momentum for employing effective security measures and making applications secure from attacks. [9]

Gollman also says that some think that computer security is "like rocket science" but that this statement shouldn’t frighten you off. If you are given the chance to implement security in a systematic way, a disciplined approach to software development and a good understanding of a few essential security principles will carry you a long way. However, you certainly will struggle if you add on security to an already complex system as an afterthought.

In a recently released Swedish book about IT-security the authors define the term as "(realistically and economically defensible) security that is about IT". Not just protecting the technology but also the information that the technology handle. The book also gives examples why it’s important to secure your IT-environment:

- If a vulnerability can damage the company’s reputation/brand
• If the company has a web service that can be used for fraud
• To guarantee accessibility for customers
• To guarantee protection for customers
• To be able too keep on working (avoid crashes)
• To protect valuable information
• To follow the terms of the existing agreement
• To avoid negative media attention
• To follow legislation [12]

The book *Writing secure code* [8] comments that many books covering building secure applications outline only one part of the solution; the code. Instead Lipner covers design, coding, testing and documentation since all of these aspects are important for delivering secure systems. It’s imperative that you adopt a disciplined process that incorporate them.

Microsoft has defined a set of guidelines called SD3+C; Secure by Design, Secure by Default, Secure in Deployment and Communication. With Secure by Design they mean that security issues should be considered in the architectural design, with Secure by Default that the product shall be dispatched with least privilege configuration, with Secure by Deployment that guides and tools should be provided so that the configuration adaptations can be made securely and with Communication. [21]
Chapter 4

The Security Development Lifecycle

In this section we will briefly run through the SDL process. The point is not to educate or show how all steps in the process are performed, but rather to give an overall picture of what’s included in the SDL and how the it takes on security. Each phase in section 4.2 is summarized in a checklist which will be used later in our study. If you are already familiar with the SDL process you can skip this chapter.
4.1 SDL History

In January 2002 Bill Gates sent out the renowned "Trustworthy Computing memo" [3] where he announced that the company would shift their focus from adding new features and functionality to security and privacy. Gates sent out memos every few years, but this one got extra attention and was a milestone in Microsoft’s history [15]. In the memo he stated that "Trustworthy Computing is the highest priority for all the work we are doing. We must lead the industry to a whole new level of Trustworthiness in computing". He goes on explaining the term; "Trustworthy Computing is computing that is as available, reliable and secure as electricity, water services and telephony". This was the starting point for a long-term commitment called The Trustworthy Computing initiative which was led by Microsoft’s Secure Windows Initiative (SWI) team. [9] In late 2003 and early 2004 they held a series of meetings with senior managers across Microsoft’s product development organizations. The meetings resulted in a decision to replace the ad hoc process of training, security pushes and Final Security Reviews (FSRs) with that essentially all Microsoft products must meet the requirements of a formally defined SDL. Security development is mandatory at Microsoft for software that is developed for the following uses:

- In a business environment
- To process Personally Identifiable Information (PII) or other sensitive information
- To communicate regularly over the Internet or other networks [19]

As you can see, this includes almost all software. The formal version of the SDL was named SDL Version 2.0, recognizing that many products had undergone an earlier, less formal, SDL process. The transition to SDL 2.0 was completed by July 2004. The SWI team is responsible for updating the SDL every six month. SDL 2.1 went into effect in January 2005 and 2.2 in July 2005 and so on. [9] The development, implementation and constant improvement of the SDL represents a strategic investment for Microsoft. [21] Besides creating the SDL process, the project to formalize it resulted in mandatory education of engineering personnel, metrics for product teams, making SWI the central security team for Microsoft and sorting out its role.

4.2 The SDL Process

"SDL is a software development lifecycle with security milestones and processes built into your overall software development methodology. The goal of an SDL is not only to produce more secure software, but to reduce the overall lifetime cost of software development projects due to the need for security bug fixes."

- Robert Harvey, Senior Security Consultant IOActive [5]
A typical software project can be represented with seven phases that are somewhat accustomed in the business (see figure 4.1).

![Figure 4.1. The seven phases in the development process.]

The SDL was designed as an integral part of the development process and can be applied to the existing development process at a company. Microsoft self explains it as "The SDL is a cultural as well as technological transformation that guides every part of our development cycle". [21] You can view it as a layer with requirements that you add to each stage of your development process that guides you to a more secure software. Here we give you a run through of the essentials of the method.

![Figure 4.2. The seven phases in the development process with applied SDL activities.]

### 4.2.1 Training

*"An investment in knowledge always pays the best interest."
- Benjamin Franklin*

Before the development team can start producing secure software they need to go through the initial stage named 'Education and Awareness'. If the engineers know nothing about basic security principles, common security bug types, basic secure design or security testing, there really is no reasonable chance that they will produce secure software. [9] All members of the software development team should receive appropriate training to stay informed about security basics and recent trends in security and privacy. A number of key knowledge concepts are important to successfully produce secure software. They can be divided into two main categories, basic and advanced (See figure 4.3 and 4.4). [19]
The basic security concepts are more or less mandatory in the SDL training were advanced concepts are recommended as time and resources permits (Note: Advanced concepts are not limited to the ones listed).

Making people into security experts is not the goal of the education. The core goal is simply to raise awareness, provide engineers with basic security knowledge, let them know what’s expected of them and tell them where they can go for more security related information. As an organization you can get education from the outside, or if the skill and knowledge already exist, perform in-house education.
4.2 The SDL Process

4.2.2 Requirements

Many development projects produce next versions of previous releases to accommodate security issues, instead the principle of secure system development is to consider security "from the ground up". The best opportunity to fit in security in the development lifecycle is in the requirement phase. [10]

There are two stages in the SDL requirement phase; project inspection and cost analysis. Project inspection is the kick off of the project. Whether it’s a new version, new iteration or a brand new product it’s important to line up all the security issues correctly. [9] The first course of action is to determine whether the product that is about to be produced is covered by the SDL. Products that meet any of the listed criteria on the check list in chapter 4.1 should be subjected for the SDL.

The next stage is to assign a security advisor. The security advisor serves as a point of contact between the security team and the development team. [10] One person from the development team will be the security contact for the team and handle the communication with the security advisor. The security advisor can be either a person from the development team or one from the security team. This person is required to be skilled in both security and project management. The goal of the security advisor is to guide the development team through the SDL process with the aim of successfully completing the FSR (see chapter 4.2.6). The tasks that the security advisor is responsible for can be summed up to these:

- Acting as a point of contact between the development team and the security team
- Holding a SDL kick-off meeting for the development team
- Holding design and threat model reviews with the development team
- Analyzing and tracking security and privacy related bugs
- Acting as a security sounding board for the development team
- Preparing the development team for the FSR
- Working with the reactive security team [9]

The next step is to identify the team or individual that is responsible for tracking and managing the security for the product. This team or individual does not have sole responsibility for ensuring that the software release is secure, but instead is responsible for coordinating and communicating the status of any security issues. In smaller product groups, a single program manager might take this role. [10] To clarify the dependencies between the different contact persons and teams you can look at figure 4.5.
When developing secure software it’s important to keep track of bugs in a bug tracking database. The SDL has requirements on how the bugs should be represented in the database. [9] It’s important to determine a bug bar where you define the criteria of which bugs should be fixed before release. The bug bar must be approved by the security advisor. [19]

That concludes project inspection, as mentioned above the next step in the SDL requirement phase is the cost analysis were you perform a Security Risk Assessment (SRA) as well as determining the projects privacy impact rating. The purpose of the SRA is to clarify the level of effort required to fulfill the SDL requirements. During the SRA the following assessments must be included:

- What portion of the project will require threat models before release
- What portion of the project will require security design reviews
- What portion of the project will require penetration testing
- The scope of fuzz testing1 [10]

The privacy impact rating measures the sensitivity of the data that the software will process from a privacy point of view. This is a slightly simpler than performing a SRA because you only need to determine which out of three policy values the software belongs to.

- **P1 — High privacy risk:** the feature, product, or service stores or transfers PII or error reports; monitors the user with an ongoing transfer of anonymous data; changes settings or file type associations; or installs software.
- **P2 — Moderate privacy risk:** the sole behavior that affects privacy in

---

1See chapter 4.2.5 for a short description of fuzz testing
4.2 The SDL Process

the feature, product, or service, is a one-time user-initiated anonymous data transfer.

- **P3 — Low privacy risk:** no behaviors exist within the feature, product, or service that affect privacy. No anonymous or personal data is transferred, no PII is stored on the machine, no settings are changed on the user’s behalf, and no software is installed. [9]

Product teams must complete only the work that is relevant to their privacy impact rating. [19] No requirement elicitation is performed in the SDL requirement phase, the security activities in each phase are pre-defined requirements.

![Checklist – Requirement Phase](image)

4.2.3 Design

In the design phase you identify the overall requirements and structure for the software and specify the functionality of the product. It’s also in this phase that you will have the best opportunity to influence security in the design specifications. [10] In this part of the SDL process you will build a plan for how the project will work through the rest of the cycle. [19] Howard and Lipner describes the SDL design phase in two stages; define and follow design best practices and risk analysis. [9]

Define and follow design best practices focus on a "good security hygiene". Design best practices involves following a set of security design principles. Examples of security design principles are:

- **Economy of mechanism:** keep the code and design simple and small. The more complex software, the greater the likelihood of bugs in the code. When the code is small, less can go wrong.

- **Fail-safe defaults:** the default action for any request should be to deny the action. Thus if the user request fails, the system remains secure.

- **Least privilege:** operate with the lowest level of privilege necessary to perform the required task. [9]
Another important best practice for the design phase is reducing a software products attack surface\(^2\). Writing secure and high quality code is not enough to make the software secure; even if the code is perfect it’s only perfect by today’s standards. New vulnerabilities and exploits are found every day. By performing an Attack Surface Analysis (ASA) you document all code, interfaces, services and protocols that are available to all users. After the attack surface is known you can perform an Attack Surface Reduction (ASR). When performing an ASR there are three main questions you can ask yourself about the elements from the ASA, "Is this feature really that important?", "Who needs access to the functionality and from where?" and "Is the feature executing with lowest set of privileges?". At a high level ASR strives to reduce the following:

- Code that executes by default
- The scope of who can access the code
- The scope of which identities can access code
- Privilege of the code [9]

The next stage in the design phase is risk analysis. Threat modeling is used to understand potential security threats to the system. It’s performed at a component-by-component level, determining risks and threats to all assets that the software must manage and establishing appropriate countermeasures to mitigate the risks. Either with a security feature or proper functioning of the software. [19] Threat modeling also helps companies to manage software risks and provide the ability to translate technical risk to businesses impact. According to Lipner and Howard, if they had to choose only one thing to improve software security, threat modeling would be it. The reason being; when performed correctly, threat modeling occurs early in the project lifecycle and can be used to find security design issues before code is committed. This can lead to significant cost savings (see table below). [9]

<table>
<thead>
<tr>
<th>Defect Introduction Point</th>
<th>Defects Found During Requirements</th>
<th>Defects Found During Architecture</th>
<th>Defects Found During Construction</th>
<th>Defects Found During Test</th>
<th>Defects Found After Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>1</td>
<td>3</td>
<td>5-10</td>
<td>10</td>
<td>10-100</td>
</tr>
<tr>
<td>Architecture</td>
<td>None</td>
<td>1</td>
<td>10</td>
<td>15</td>
<td>25-100</td>
</tr>
<tr>
<td>Construction</td>
<td>None</td>
<td>None</td>
<td>1</td>
<td>10</td>
<td>10-25</td>
</tr>
</tbody>
</table>

**Figure 4.6.** Table over the relative cost of fixing a bug in a certain phase.

\(^2\)Attack surface is the sum of all code and functionality accessible to users and potential attackers.
4.2 The SDL Process

4.2.4 Implementation

Even though security is important to focus on early in the development lifecycle we can’t forget about it in the later phases. During the implementation phase the SDL focuses on two stages; create documentation and tools for users that address security and privacy; establish and follow best practices for development. [19]

Why create documentation and tools for users? Is it not enough that the software is secure? As we mention in chapter 3 all software that is released should be secure by design, in its default configuration and in deployment. However, people use programs differently and not everyone uses a program in its default configuration. [9] To ensure that the security implemented in the software works as it should after release, enough information must be provided to the users so that they can make informed decisions about how to deploy a program securely. Documentation is delivered to users as security manuals and tools as security wizards. [19]

As in the design phase where we followed design best practices, here we follow software coding best practices. The SDL process mandates specific coding practices and back them up later during the verification phase to make sure they have been applied correctly. [9] The details in secure coding changes all the time, new vulnerabilities and exploits are found regularly. For more specific and up to date information on SDL coding best practices visit the msdn homepage3.

Here follows the current mandatory coding practices for SDL:

- **Use the latest compiler and supporting tool versions:** it’s important to use the right type of compiler when developing software to get built in defenses. This defensive code is generated by the compiler and not produced by the developer.

- **Use defenses added by the compiler:** the SDL process specifics the defensive options that must be used.

- **Use secure-code analysis tools:** there exist many code analysis tools to automate search for bugs, which one you use is not that important. However, you have to remember that automated tools are not a one way ticket to secure software. Code analysis tools don’t detect design errors, can miss bugs and create false positives. Manual code reviews are needed to compensate the lacks of automated analysis.

---

- **Do not use banned functions:** there exist a lot of functions that, although fine 20 years ago, are simply not secure by today’s standards.

- **Use a secure coding checklist:** making secure coding checklists to be used when checking in code to the software is a great help to see if other best practices have been upheld. [19]

As mentioned before the coding best practices are not limited to these, continuously checking for updates and new techniques is recommended. This links back to the yearly education for developers in the training phase of the SDL.

---

### 4.2.5 Verification

In this phase most of the implementation is complete and we have something that is functional, it’s now time to verify all the hard work that has been done in the previous phases. This is done by security and privacy testing together with an over all security push explained below.

The testing can be divided into two general areas; testing issues, like buffer overruns; and testing the confidentiality, integrity and availability in the functionality of the design. [19] Security testing will not introduce security into the product but only verify if it is or is not there. The SDL uses the following steps when performing security testing:

- **Fuzz testing:** this means creating malformed data and feeding it to the system that’s being tested to see how it reacts. There exist different ways of fuzzing and how it should be used depends on what’s tested.

- **Penetration testing:** this type of tests is designed to find vulnerabilities in information systems by simulating an attack from a malicious source.

- **Run-time verification:** run the application on a regular basis to find bugs and flaws in the execution.

The threat modeling document is useful during testing. Some times functionality and implementation changes late in the development lifecycle, review and update threat models when needed. The same thing goes for the attack surface, which is very useful during penetration testing. [9]

The SDL security push is introduced in the later parts of the verification phase, its focus is to uncover any unexpected changes that might have occurred during
4.2 The SDL Process

development, improve security in legacy code⁴ and identify any remaining vulnerabilities. This is a project wide effort where you look at the entire system, performing code reviews and security tests, updating threat models and verifying security documentation. The SDL also includes a training stage before the security push to get everyone that’s involved in the development team up to the same knowledge level.

4.2.6 Release

When the project draws to an end and it’s time for release, the SDL emphasis on two things that are important from a security perspective; the FSR and security response planing.

The FSR should be an independent review and not performed by the product team. Instead it’s conducted by the central security team of the organization. The FSR is a review verifying that the product team has followed the SDL correctly. If one would sum up the goal of the FSR one could say it’s the answer to the following question: "From a security and privacy perspective, is the product ready to ship to customers?". [10] The FSR starts of with a short questionnaire for the development team to help out the reviewers with the following tasks:

- **Threat model review**: one of the cornerstones of the SDL is threat modeling and it’s important to verify that it’s up to date and correct. It’s also important to see that all appropriate mitigations are in place.

- **Unfixed security bugs review**: in this part of the FSR you review all bugs listed during the development as "don’t fix" bugs to see if they really don’t need to be fixed.

- **Tool-use validation**: the FSR team will verify that all appropriate tools have been used during the development.

If something comes up during the FSR that is against the SDL the security team and the security advisor will either come to a compromise or go back to the root of the problem. The FSR is not only a pass or fail exercise and it’s not constructed to find all remaining vulnerabilities. Rather its purpose is to give management an over all picture of how secure the product will be after release. [9]

---

⁴The term legacy code refer to a chunk of code or a program that still is apart of the system but was developed along time ago.
Even if the SDL process was followed to the letter mistakes occur and new unpredictable events can always happen. No software is 100 percent secure so a response plan, describing how your team should react to a security issue, is always needed. How to make a response plan is out of the scope for this thesis. [19]

4.2.7 Response

This phase is not much more than a reminder to follow the pre-defined response plan made in the release phase if a security issue would occur. What’s also important is to keep the plan up to date and change it if needed. [19]

4.3 SDL-Agile

The SDL process described earlier is a classic waterfall model and can be a bit too heavy from a agile point of view. Agile development is wide spread today and acknowledged methods that are frequently used both in small and big organizations. One problem with agile methods is that they are not designed for security, and that is why Microsoft has presented an alternative way of using the classic SDL process; SDL-Agile. Two main changes have been made to the SDL process so it can be adopted by agile methods. First, SDL additions to agile processes must be as slim as possible for every feature developed so the workload doesn’t get out of hand. Second, the development phases that are described in the classic SDL don’t apply to agile methods and therefore need to be reorganized.

If you are unfamiliar with the agile method we can recommend agilealliance.org for more information. What you need to know about to apply the SDL to agile methods is the sprint. A sprint is a short period of time (usually 15 to 60 days) within which a set of features are designed, developed, tested and then potentially delivered to customers. The list of features to add to a product is called the product backlog, and the set of features which are to be processed in every sprint
is called the sprint backlog. Since the SDL is divided into requirements that can be represented as tasks they can be added as features in the product and sprint backlog. We still have the problem that the SDL contains too many requirements to be fulfilled during a sprint. SDL-Agile resolves this by dividing all requirements into three categories.

- **Every sprint requirements**: These SDL requirements are so essential that no software should ever be released without fulfilling them. Whether the sprint is two weeks or two months long, all these requirements must be completed in every sprint.

- **Bucket requirements**: Requirements belonging to this section must be performed on a regular basis over the lifetime of the project, but don’t need to be completed in every sprint. The requirements are divided into three subcategories called buckets (see figure 4.7). All bucket requirements don’t need to be met in every sprint, instead the product team must complete one task from each bucket in every sprint. It’s up to the product team to decide which task from each bucket they need to address, however all the bucket requirements need to be completed over the whole lifecycle within six months.

  ![Figure 4.7. The bucket requirements are divided into the categories; Verification Tasks, Design Review and Response Planning](image)

- **One-time requirements**: These are the "once-per project tasks" that don’t need to be repeated after they are completed. Unfortunately all these requirements can’t be fit into one sprint. SDL-Agile addresses this with a grace period for each requirement reaching from one month to one year depending on the complexity of the requirement. For example, choosing a security advisor is a one time task and has a grace period of one month after project start.

A list of requirements and which category they belong to can be found in appendix A. The SDL-Agile process can be represented with figure 4.8.
Some side notes worth thinking about are threat modeling, the security push and the FSR. Threat modeling is one of the every sprint requirements but is hard to automate and can take a lot of time. It’s important to note that only new features and changes need to be threat modeled in the current sprint.

The security push is managed as a spike\(^5\) and if serious vulnerabilities are found that need to be fixed at once, a more thorough spike will be launched to find more security vulnerabilities. Finally the FSR is not as strict as the one in the classic SDL. The security advisor only needs to review the following at the end of every sprint:

- All every-sprint requirements have been completed, or exceptions for those requirements have been granted.
- At least one requirement from each bucket requirement category has been completed (or an exception has been granted for that bucket).

\(^5\)A spike is a story that cannot be estimated until a development team runs a timeboxed investigation.
• No bucket requirement has gone more than six months without being completed (or an exception has been granted).

• No one-time requirements have exceeded their grace period deadline (or exceptions have been granted).

• No security vulnerabilities are open that fall above the designated severity threshold (that is, the security bug bar). [20]

4.4 SDL and Management

A point that the authors of the book *The Security Development Lifecycle* [9] bring up repeatedly is that one of the most important things about introducing the SDL is that management is backing up the commitment; ‘Commitment for Success’ they call it. Executives and managers play a vital role in the implementation of the SDL in a software development organization. Howard and Lipner also mean that it’s very important that managers understand the SDLs impact on the software they produce and the expectations that SDL places on the stakeholders in their organization. They focus on the role of managers in making the SDL succeed; what the manager or executive must do to ensure that her or his team can build more secure software. It requires strong commitment of senior managers to prioritize security over other factors such as time to market and compatibility to older, less secure software versions. If the management wants to know the impact of the SDL project cost and schedule, monitoring the deliverables and activities associated with each phase in the SDL can give them a clear idea whether the project is on track and how much the SDL is costing. Also, tracking external measures such as customer satisfaction with security and the rate of security incidents affecting products and services, can give managers a similar understanding of the benefits of implementing the SDL. [9]

One thing management usually understands is money. To show that security in the end will cause the organization to get a profit will certainly help introduce security in development. In September 2009 Microsoft released the white paper *Microsoft SDL: Return-on-Investment* [18]. The paper is trying to show how investing in security upfront on a project will give a bigger return of investment (ROI) than only applying it at the end, and how an organization can calculate the ROI for security processes. The main argument for introducing security early on, is based on a study by the National Institute for Standards and Technology which says that a bug discovered in the verification phase can cost up to 30 times more than a bug discovered in the design phase.
Furthermore they recommend the following six activities when starting a new software security effort to provide a good ROI.

- Plan carefully before you work to structure your process. The plan should not only cover the basic execution of the process but also include metrics to evaluate the effectiveness of actions before they are taken.
- Start at the beginning to leverage structural efficiencies.
- Select pilot projects to test out your formed security process. Pay attention to risk management and pick a project with a security risk. This will increase the effectiveness of your process and justify future efforts.
- Build appropriate standards and policy.
- Create an incident response plan.
- Get expert help when appropriate. Consultants or some existing programs like the SDL Optimization Model.

The actual numbers on the ROI is different depending on the organization and what type of security activities that are performed. From the recommendations described above, the paper presents a calculated example of the ROI at a small development organization with around 50 employees. The end result was a ROI of $350k for some initial activities. The activities, calculation and matrices used can be seen in appendix B. [18]

4.5 Does the SDL work?

Microsoft has adopted the SDL because they have a goal to create more secure software and reduce customer pain. In the introduction of the book on SDL they
4.5 Does the SDL work?  

say; it’s not just a theory, SDL works. They go on saying that SDL is based on real-world experience. The ultimate test of the SDL is the extent to which it can reduce the number and severity of vulnerabilities in software. [9] In order to measure how these goals were met, security experts analyzed public vulnerability counts in "pre-SDL" and "post-SDL" versions of the same product. [22] Here we give examples of what Microsoft use as proof to the positive effect of using the SDL.

Windows Server 2003
The first Microsoft product to practice most (but not all) of the SDL processes was the operating system Windows Server 2003 (WS2003). Here we can see a chart (figure 4.10) comparing the number of security bulletins issued within the year after release between WS2003 and Windows 2000 server which was developed without the SDL and is the most recent server operating system before WS2003. [10] As you can see there is more than a 60 percent decrease in security bulletins.

![Figure 4.10. Windows pre- and post-SL, critical and important security bulletins.](image)

Windows XP vs. Windows Vista
Windows Vista was the first major software release done entirely with the SDL-process. [5] One year after release 66 vulnerabilities had been reported, which can be compared to the pre-SDL OS Windows XP that had 119 vulnerabilities reported one year after release. [25] Below is a chart of these numbers, including competing OSs vulnerability reports. Note that there is a 45 percent reduction of vulnerabilities from XP to Vista.
"Conventional wisdom[...], tends to suggest that the more lines of code you have the more bugs you have. That might very well be true, and Windows Vista is certainly larger than Windows XP SP2; yet right now, we are on track for an approximately 50% reduction in vulnerabilities compared to Windows XP SP2[...] why are we seeing a reduction in vulnerabilities? Simple: the SDL!"


Furthermore, in the first year after the release of Windows Vista, Microsoft released 17 patches, fixing a total of 36 security vulnerabilities. The shift to the predictable monthly patch release policy resulted in that there were 9 days in the year when security updates were released.

In the first year after the release of Windows XP, Microsoft released 30 patches,
4.5 Does the SDL work?

fixing a total of 65 vulnerabilities, on 26 different days throughout the year. [11]

![Figure 4.13. Patches released one year after Windows XP was released](image)

As you can see the result is 9 vs. 26 patches! In conclusion the SDL combined with the predictable monthly patch release policy resulted in more than 65 percent less patches which effected all companys that use Vista, saving them a lot of time and money on patching activities. [1]

**SQL Server 2000 vs. SQL Server 2005**

The effect has been obvious also for other Microsoft products. 36 months after

![Figure 4.14. Vulnerabilities found after release in Microsoft SQL server](image)

the release of pre-SDL SQL Server 2000, Microsoft reported 34 vulnerabilities, as compared to only 3 for post-SDL SQL Server 2005. That is a 91 percent decrease in vulnerabilitys. [25] Since the third update to Microsoft’s SQL database server was released, the software has had zero vulnerabilities in 24 months.

**ISS**

Microsoft’s Web server Internet Information Services 6 (IIS6) is called the poster
child for the SDL. IIS6 has had one security vulnerability since it was shipped, and it was for a feature that wasn’t even turned on by default. [14]

“*We actually consider Microsoft to be leading the software [industry] now in improvements in their security development lifecycle, Microsoft is not a punching bag for security any more*”
- John Pescatore, Gartner, former vocal Microsoft IIS critic [9]

4.6 Spreading the word

Today the main source of information about the SDL is its dedicated homepage\(^6\). There you can find whitepapers, tools and FAQ on the SDL and everything is for free. Microsoft also has a dedicated blog for the SDL, "SDL Blog\(^7\)" where you can get the most up-to-date ideas and thoughts from the SDL team members at Microsoft. One of the authors on the book on SDL, Michael Howard, also has a blog where you can read all about security in software development. The book he and Steve Lipner wrote, *The Security Development Lifecycle: A process for Developing Demonstrably More Secure Software* [9], is another source of knowledge and Howard’s book *Writing Secure Code* [8] is recommended as further reading and a well acknowledged book by the community.

The Optimization Model
The Optimization Model is a framework for making a gradual, consistent and cost-effective implementation of the SDL for development organizations that are adopting the process. The model includes an overview and how to use it, a questionnaire for mapping the current process to the SDL maturity levels and detailed guidance on how to move up in the SDL maturity levels.

![Figure 4.15. An overview of the Optimization Model.](http://microsoft.com/sdl)

\(^6\)http://microsoft.com/sdl
\(^7\)http://blogs.msdn.com/sdl/
The model enables development managers and IT policy-makers to:

- Assess the state of the security of their development organization by using a scale of four maturity levels

![Figure 4.16. The maturity levels in the Optimization Model.]

- Create a practical vision and roadmap for moving up the SDL maturity levels in each of the five software development capability areas to improve the state of security and reduce customer risk

![Figure 4.17. The capability areas of a software development process.]

- Outline practical and cost-effective activities in each of the five capability areas to assist with budgeting, planning and staffing efforts associated with software development [19]

**SDL Pro Network**

To help development organizations adopt the SDL and embed security and privacy into their software and development culture, Microsoft created the SDL Pro Network which was launched in November of 2008. The SDL Pro Network consists of training and consulting companies that specialize in application security and security training and have substantial experience and expertise with the methodology and technologies of the SDL. The Network runs in a pilot phase the first year of operation and during this time, Microsoft and the additional member companies will evaluate how to best expand the program to others in the industry. The objectives of the SDL Pro Network are to:
• Provide guidance and support to small and large development organizations in implementing the SDL into their environments and protecting their customers.

• Give feedback on customer needs and implementation practices to continuously improve the SDL. [23]

"So, what are we hoping to gain by creating a network of security consulting and training experts to work with customers who want to implement the SDL? Generally speaking, this question has a two-part answer: First, Microsoft is, and always will be a partner-driven company [...]. Second, even though there are talented folks in the Microsoft Services organization, it’s clear that we will need help from our partners to scale to meet the demand."

-David Ladd [13]

4.7 SDL and the Software Business

The SDL book recommends that all software products that have a considerable amount of users should work with improving their software security. The reason is that the sheer cost of applying security updates makes it worth getting security, privacy and reliability right the first time. You should get it right early in the process rather than putting the burden to apply updates on your customers. Another reason, is that every security vulnerability puts your customers at risk of an attack or exploitation. They go on stating that if your software is a business-critical application, improved security should be an easy sell because of the business impact of a failed system. The SDL is not free, it costs time, money and effort to implement, but the upfront benefits far outweigh the costs. [9]

They also challenge all ISVs to change their software development process; "If you are not implementing a process similar to SDL, the process you have now simply do not create more secure software products. It’s time to admit this and do something about it. Your customers demand it." This goes especially out to software vendors with more than 100,000 customers. To follow up their own challenge they add that Microsoft has already taken it on, in the form of the SDL permeating the company. Ending the argumentation with; "You must do likewise, or attackers will smell blood and the competition that offers products that are more secure than yours will take sales from you. Rebuilding customer trust and goodwill will be difficult at best. We say this from painful experience." [9]

To encourage in-house developers to create more secure software the authors of the book point out reduction in privacy and reliability exposure. Especially the privacy exposure risk is something that senior management and risk managers understands, they say. They recognize that it’s harder to motivate small companies because even small amounts of security costs time and money. But implementing more secure software up front saves you time and money in the long run. Another problem with small development companies that they acknowledge, is that they often have a lot of
pride and ego tied up in their code (commonly known as the "ugly baby syndrome") and that can be solved with looking at security as a measure of quality. [9]

"It’s fair to say most people don’t mind doing hard work; they just hate reworking" - Howard and Lipner [9]

As stated earlier in the beginning of the SDL chapter (4.1), Michael Howard writes in a presentation about Pragmatic Trustworthy Computing, that the key goal is to make computing so safe and reliable that people simply take it for granted, just as they take electricity and telephones for granted today. He then details what this commitment means:

- A top priority for Microsoft and a cultural shift we are totally committed to
- A lengthy journey (at least a decade)
- Needs the commitment of the entire computer industry (software, hardware, ISPs, etc) [7]

It’s the last item that is important to notice in this chapter; that Microsoft has recognized the need for commitment of the industry and started acting on it as a part of the Trustworthy Computing initiative. A trend that they are seeing urges the commitment further; since the SDL has significantly improved the security of Microsoft flagship products, attacks are shifting to the application layer (see figure 4.18). This makes it critical for application developers to adopt the SDL. [23]

![Figure 4.18](image-url) The graph presents the distribution of attacks over OSs, browsers and applications.

"Microsoft has put its first-hand knowledge of dealing with buggy software to very good use by releasing the Trustworthy Computing Security Development Lifecycle"

---

8 Internet Service Providers
Many have recognized Microsoft's effort and are encouraging it. Jon Oltsik, Principal Analyst at Enterprise Strategy Group (ESG), expresses this in his brief called "Microsoft Takes its Security Development Lifecycle to the Rest of the World" with a part called "The world needs SDL". Here he emphasizes the importance to note that Microsoft won't gain a big payoff with its SDL externalization. He answers the question on why Microsoft continue to pursue this strategy with that they realizes that a lot of insecure software sits on top of Windows and regardless of who writes the code, Microsoft will somehow take the blame. Oltsik goes on saying that Microsoft deserves high praise for creating, formalizing, and improving the SDL as it has led to better software for the masses. He sees the initiative as a natural progression and thinks that organizations that embrace the SDL can improve the security of their code while decreasing the cost of maintenance and security operations which he calles a true win-win. [25]

"The only issue here is artificial-SDL is synonymous with Microsoft. In spite of its independent nature, the market may somehow perceive SDL as a proprietary agenda rather than well formulated training, modeling, processes, and testing."

- Jon Oltsik, Principal Analyst ESG [25]

Microsoft has tried to meet the need for software security in the industry by publishing papers, books and free information on their homepage to share their experience across the software industry. [10] Besides this effort they have also put forward the importance of federal privacy legislation to the U.S. Congressional Internet Caucus through a speech made to them by Senior Vice President and General Counsel Brad Smith in November of 2005. The legislation would, in Microsoft view, not only better coordinate privacy protection within the United States, but also better align U.S. protection with those offered by countries around the world. Microsoft is also continuing to use the legal system to address online safety threats. In 2005 they took a number of legal and enforcement actions, such as lawsuits and participation in investigations and arrests, to get spammers, phishers, and virus launchers off the Web for good. [24]

"Microsoft will continue to push on all fronts towards our goal of trustworthy computing, but it will take a strong and focused effort from industry partners, government and law enforcement if we are going to reach our long-term goals of providing a more safe and secure computing experience for every one of our customers."

- Mike Nash, Corporate VP of Microsoft's Security Technology Unit [16]

Microsoft also continues to be active in industry organizations that work toward trustworthy computing goals, including TRUSTe, the Global Infrastructure Alliance for Internet Safety, and the Anti-Phishing Working Group. [24]
Chapter 5

Security Development in the Software Business

In this section we will present the collected data from our interviews. We have performed them on three selected groups which we call; MS-profiles, who are evangelists at Microsoft’s Sweden office; Security experts, who work with security as consultants and ISVs, who is our selected target group. The questions used in our survey can be found in Appendix C.
5.1 The View on Security at Microsoft

We have interviewed five different MS-profiles and most of them have regular contact with our target group. They intermediate between ISVs and experts in different fields including security. A summary of the results from the interviews is presented in figure 5.1, a more detailed description can be found in the following subsections. The results are rated as low, medium or high and represents how MS-profiles thinks of the current situation at mid-sized ISVs.

<table>
<thead>
<tr>
<th>Existing process</th>
<th>LOW</th>
<th>Information flow</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing security process</td>
<td>LOW</td>
<td>Education</td>
<td>LOW</td>
</tr>
<tr>
<td>Security awareness</td>
<td>LOW</td>
<td>Feedback</td>
<td>LOW</td>
</tr>
<tr>
<td>Maturity</td>
<td>LOW</td>
<td>SDL awareness*</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Legislation</td>
<td>LOW</td>
<td>Estimated vulnerability level</td>
<td>HIGH</td>
</tr>
<tr>
<td>Future security awareness</td>
<td>MEDIUM</td>
<td>How important is security?</td>
<td>HIGH</td>
</tr>
<tr>
<td>Knowledge</td>
<td>LOW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 5.1. Results from our interviews with the Microsoft profiles. *How aware the MS-profiles are of the SDL

5.1.1 Development at the ISVs

According to the profiles, today’s ISVs are focusing on new features, flashy interfaces and sales are holding back on security because it doesn’t sell. The ISVs are starting to realize that having security as a consideration from the outset is more cost effective than having to pour resources in after the fact to try to patch things up. Secure solutions can be big and complex and cause problems with basic features; the secure implementation can be twice the size of the basic one. The knowledge and maturity level in the business is too low, the profiles think. There is a demand but it varies. The degree that the company focuses on security depends on the maturity level and what security requirements the project has. The profiles believe that the ones that are affected by legislation care more and that there is a demand for methods for secure development when the nature of the project leads to strong security requirements. The maturity level is starting to rise, but there are large gaps between how ready companies are to invest and most of them aren’t. There is also a polarization in the business, those who are good at security are really good, but the majority do the best they can.

Small companies only objective is to make money and they have practical programmers that "hack like hell". The opinion of the profiles is that customers are not prepared to pay the extra cost necessary to have security implemented. To
follow standards is secondary, design for operations is more common. There is information about secure development but there is no interest and the knowledge is still low. it’s too easy to do something wrong. Companies are trying to find short cuts because they can’t take all the steps in a security development process, but taking short cuts often requires more knowledge than just following a method.

5.1.2 Information Flow and Influence

The perception on how to spread information about secure development and the SDL was quite unanimous amongst the evangelists at Microsoft Sweden. They want to be able to show the benefits of applying an entire security process at smaller organizations instead of using security patches and features. Another thing they think is needed is to make the SDL more human and easier to understand than it is at the moment. It’s hard to read about security and get a good understanding on the subject. Strong role models and mentoring have worked best so far in increasing the security awareness in the business. Finding a really good case, where the SDL has been used and the positive results can be linked back to the process, would be a good way of spreading the word about security. Another way is to show existing vulnerabilities and possible attacks on products. Showing how much more secure Microsoft products have become and the profit they make on security is yet another way. ISVs are easier to influence than consulting businesses, they often have their own development process where you can integrate security, while consulting businesses don’t. What’s important in the end is that others learn from the mistakes that have been made and from the knowledge that is out there.

The opinion whether Microsoft have a responsibility of spreading their knowledge and experience to the rest of the business differs. Some say that because Microsoft is one of the leading companies that produce secure software and has a big influence in the business, of course they have a responsibility to the rest of the community. Others say of course Microsoft should help, but the sole responsibility should not rest on them but on independent organizations. Either way it’s difficult for Microsoft to tell other organizations what to do and the good intentions of spreading the information gets stuck some where in the political wheels.

The Microsoft Sweden office wants to spread the knowledge to universities but don’t know how, at the moment their focus is marketing and to spread knowledge is not prioritized.

5.1.3 The Future

People 'on the ground' that are committed to security are needed. The SDL should be Microsoft Solution Framework packaged and not treated as a separate process. The customers need to make requirements for security to affect the business. The most important thing isn’t that they use the SDL, it’s that they use any type of method.
Nitched companies that think about security hopefully have better customer satisfaction than those who don’t and hopefully you can recoup money if you have good security awareness.

The profiles also say, that hopefully we will talk about security in a more human way; you don’t need to be a hacker to talk about it, normal intelligence is enough.

5.1.4 The SDL

The knowledge about the SDL and security at Microsoft Sweden is decent, they know about solutions, products and can redirect to partners that provide solutions and guidance, for example on threat modeling. The problem is resources, they are too few in the Sweden office to cover all areas within Microsoft and unfortunately the SDL is one area that’s not prioritized in Sweden. They do get training in security, a few times a year they get to go to Redmond and get information of the newest updates. The SDL process has shifted the recourse allocation a bit at Microsoft, fewer recourses are needed for patches and can be spent on research instead.

5.2 The View on Security from the Experts

We have interviewed three different security experts. Two of them are security consultants and work with security processes on an everyday basis at companies from our target group and one is program manager at a SDL-practicing development project. You can see a summary of how experts view development at ISVs in figure 5.2.

<table>
<thead>
<tr>
<th>Existing process</th>
<th>MEDIUM</th>
<th>Information flow</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing security process</td>
<td>LOW</td>
<td>Education</td>
<td>LOW</td>
</tr>
<tr>
<td>Security awareness</td>
<td>LOW</td>
<td>Feedback</td>
<td>LOW</td>
</tr>
<tr>
<td>Maturity</td>
<td>LOW</td>
<td>SDL awareness*</td>
<td>HIGH</td>
</tr>
<tr>
<td>Legislation</td>
<td>MEDIUM</td>
<td>Estimated vulnerability level</td>
<td>HIGH</td>
</tr>
<tr>
<td>Future security awareness</td>
<td>MEDIUM</td>
<td>How important is security?</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Knowledge</td>
<td>LOW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.2. Results from our interviews with the security experts. *How aware the experts are of the SDL
5.2 The View on Security from the Experts

5.2.1 Development at the ISVs

The experts agree on that the threats to software today have moved from OSs to the application layer and the attackers are getting smarter and more organized. Application security is a fairly new subject, only 3 or 4 years old, and the application business is not aware of the rising threat. They believe that most mid-sized companies today are using agile methods in their development and functionality is in focus. The experts think that the maturity level for security in the business is very low or practically non-existing, if security is mentioned or thought of it’s in the sense of firewalls, encryption, SSL and other patch related security features. If there exist any functional security requirements in the project they are upheld, but they are more likely to be about infrastructure security than about building more secure products.

Security consultants are hired by companies that are more or less aware of their threats; banks, government, the county etc. They are brought in late in the development lifecycle, often in the quality assurance phase, and are mostly expected to perform testing and code analysis. The code analysis is done manually or with open source tools. The security experts’ opinion is that commercial analysis tools are better suited for the task then open source ones. The problem is that the commercial tools are too expensive for the security consultants. They need to convince the company to buy them and it’s usually too expensive for them also.

One of the security experts we talked to had a good metaphor for the security awareness in today’s software business:

"The software business today is like the industry was back in the 70’s. They used all means necessary to make a profit and did not care about toxic waste and pollution. The ‘clean up’ has been made retrospectively partly by making the industry processes more environmental friendly. It didn’t happen over night and the industry had to get a stable production first. The same can be said for the software business."

The experts are starting to see a slow change in the business. More and more organizations are seeing the benefits with security and the end customers are starting to get an interest in it.

5.2.2 Information Flow and Influence

The security awareness depends on what business you’re in, banks handle security and so does insurance and online betting companies. Online betting is especially a target for organized crime and fraud, so the companies behind the sites are well aware of security risks. They have a tougher job than banks since the transactions are made 24 hours a day, not like banks who can only make them by night and thereby avoid risks. One expert with experience on the subject thinks that the county councils need a better security awareness; today they don’t even have their own testing environment for the software that is produced for them.

Consultants are often a part of the secure development, as they are hired to improve
the development process. It’s as easy to do things right as to do them wrong is the experts’ opinion, and it takes as much time to write secure code as to write insecure code. Some experts say that the managers have no influence on security, only internal heroes who have a vast security interest can affect the company. The heroes are the ones you go to if you have a question about security, and they are the ones whose advise you seek and who influence if a security change is needed. Other experts say that you always need to convince management to include security; management wants to understand risks and you need to calculate risks to be effective and to get management approval. Management understands security easier if it’s translated into money or loss of reputation. Security is a competitive advantage and it’s a myth that you get exposed to more risks if you announce that you use a secure development process. On the other hand, if you say that you are "Hacker-proof", it will trigger threats.

The security awareness has changed just three years ago the common attitude was "what’s the probability that we will be hacked?!" (assuming that it’s low) and security was taken lightly. They think that case studies and success stories within the own business affect and are the most valuable. Other stories are not entirely uninteresting and can also work as eye openers and indicators. General cases are really important. For example, in medical care they use cases that are anonymous that others can learn from, the experts think the same should be done in the software business. According to them, attacks must occur before companies will include security measures, either on their own products or they must read about occurrences at similar companies in the media. The awareness could also rise by raising the question at conferences and seminars. The catalysts can be concluded to security consultants influence, attacks and conferences including the subject. These, in turn, will lead to a chain reaction where more will be published about security. The attention in media will lead to more discussions, which will lead to that security is included in education, which in turn will lead to that security development processes will be brought to attention and introduced at companies.

The experts think that technical educations plays a substantial role in the influence on security. Today development isn’t seen from a security point of view at the universities, the students need to learn what design level to choose to provide security. It’s up to an individual developer to know what’s good design and for companies to see the benifits in the long run and not just the current cost for development. Companies today don’t have the security knowledge in-house and consultants (i.e. our experts) are hired to educate the co-workers. One recommendation from the experts is to get the book "Writing Secure Code" [8] and have it as mandatory reading for all co-workers who come in contact with software development. Another tip is to at least have one education a year to update the staff about security risks. The experts themselves attend security conferences and seek knowledge on the Internet on blogs, twitter feeds, news, statistics, white papers, online-courses etc which can be recommended to others.
5.2 The View on Security from the Experts

5.2.3 The Future

The experts believe that all Internet facing applications need high security and that the most efficient way is to use processes and proactive methods to include security. You don’t need to strictly follow a processes, it’s better to combine knowledge and tools to fit customers’ need. It’s not all up to the process; security has three ground pillars: people, processes and technology. All of these need to work together to produce secure software. A good first step to integrate security in the development is to introduce a security advisor. All organizations need someone committed to security to be able to kick start a change.

It’s important to have employees that are up to date on security, having security training during the university years is good but not enough. Regular security training together with real life experiences is needed.

Something most security experts agreed on as the 'best' security activity is threat modeling. The reason for this is; it’s 'time boxed', you can only perform it for a certain amount of time; it doesn’t require that much training to perform and it gives great result. Unfortunately it’s not used that much in practice, what’s more common is risk assessment and these two are often mixed up.

Security is not everything. Usability often contradicts security in many ways and having an unusable but very secure product will not profit sales. Even if security is extremely important in today’s software, the experts think that usability should always come first.

5.2.4 The SDL

The experts recommend that companies should introduce a security process. They think that most don’t have a clue what the SDL is, at most they know about some agile process. One way, which they themselves use, is to have the SDL as a template and adapt it to the companys needs. You can start by using OpenSAMM\(^1\) [26] or the SDL Optimization Model to learn which maturity level you have and where to start. OpenSAMM is preferred by some experts because it’s more detailed. One thing to remember is that you don’t need to use the whole SDL process. All of the experts agree that the most important activity in the SDL is threat modeling. They think threat modeling works for a novice, everyone can use it and you get alot of security from it. Code scanning is also important and one of them think that it’s even equally important to threat modeling. Further, you shouldn’t search for cross scripting, SQL-injections and the like manually, because it’s time-consuming, instead use tools. Experts are good for finding logical flaws and it’s inefficient to hire them for the activities just mentioned.

Things that the experts think is missing in the SDL is abuses cases, requirement formulation, verification (test cases for non-functional requirements) and vulnerability scanning tools for reporting statistics. Testing tools for vulnerabilities in

\(^{1}\)OpenSAMM is another guide to integrate security in development.
run-time are also needed. The experts think that Fuzz-testing is mostly necessary at companies like Microsoft for finding buffer overflows; it’s a very OS-specific measure. Improvements could also be made in the guidelines for the SDL and a better guide for introduction, having the OpenSAMM as a model, should be made. In OpenSAMM they have a questionnaire to guide you on how to get started, this could be useful in the SDL too, they think. Also, having an "SDL-light" alternative would be good. One criticism is that the SDL is not made for flat organizations like the ones we have in Sweden but especially suited for American structured companies.

The experts agree that SDL Pro Network is a very good initiative and an expansion would be a good thing. There is an interest for it in Sweden, some of the experts wanted to join right away.

5.3 The View on Security at the ISVs

To find out the reality at our target group, mid-sized ISVs, we have interviewed four representatives at four different target group companies. When possible, we have talked to the security manager.

<table>
<thead>
<tr>
<th></th>
<th>Existing process</th>
<th>Information flow</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing security process</td>
<td>Low</td>
<td>Education</td>
<td>Low</td>
</tr>
<tr>
<td>Security awareness</td>
<td>Medium</td>
<td>Feedback</td>
<td>Low</td>
</tr>
<tr>
<td>Maturity</td>
<td>Low</td>
<td>SDL awareness*</td>
<td>Low</td>
</tr>
<tr>
<td>Legislation</td>
<td>Low</td>
<td>Estimated vulnerability level</td>
<td>Low</td>
</tr>
<tr>
<td>Future security awareness</td>
<td>High</td>
<td>How important is security?</td>
<td>Medium</td>
</tr>
<tr>
<td>Knowledge</td>
<td>Low</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 5.3. Results from our interviews with the ISVs. *How aware the ISVs are of the SDL

5.3.1 Development at the ISVs

At the ISVs we have visited there has been a few variations in how development is performed, but in general the overall picture on security has been quite similar. They are all using their own variation of some existing development process and the majority is using agile methods like Scrum. The initial response to the question "do you consider security in your development?" has been either: 'No', with the defense that their products are in a closed systems or integrated in other
5.3 The View on Security at the ISVs

software. Or "Yes of course, we have firewalls, use SSL, certificates, access control, encryption and security patch related solutions." When we moved on and asked questions related to security in the development processes, most answers were "that would be good, but no, we don't use that". The main reasons for not using certain security activities has been been time, money and customer demands. Fast marketing is prioritized over security because they want a strong economical base before giving resources to security. Activities like code analysis, audit and some coding principles are in use at some of the ISVs. The only time security is in focus is when they have functional security requirements from customers. One of the ISVs mention that when they do implement security in their features or products they don’t have it as default, it’s up to the customers to turn it on. Another big problem the ISVs brings up is legacy code, their products were developed a few years back and contains a lot of code which would take a lot of time and money to refactor. This is one of the reasons why patch solutions are easier to implement.

The only laws that have been mentioned in our studies that effects security have been the audit laws\(^2\) and encryption restrictions.

All the ISVs have been aware of security and agree that it’s important and that it effects the companies reputation. Some even mention that an unsecure product is a non-functioning product while others say that security only effects the company’s reputation when something bad happens. No one have had specific security policies, some one mention that the only security policies they have is their own common sense.

5.3.2 Information Flow and Influence

More general education is given at the ISVs, either there’s a policy that you get to go to a conference once or twice a year or you have a individual plan for personal development. Some even have an internal training team where they push for security. At the conferences it’s of one’s own accord to choose which sessions to attend and if you attend any security sessions. Education is also provided to some extent by books, magazines etc. They don’t have any special or regular security education but are interested in and would like to have it. The general education level in security is the one you have after a 3-5 year computer science education at the university. Some co-workers are interested in security, mostly the ones on the framework side\(^3\), and it’s noticeable through the questions that are brought up by them.

The general notion is that the customers and users don’t have any knowledge or interest of security and there are no demands about it at the ISVs either. Mostly because it’s the customer who directs what is developed. The ISVs think that security is hidden and is hard to demonstrate for the customer, it’s easier to show interface and functionality changes. Some try to get their customers aware of

\(^2\)the term audit means an independent review and examination of records and activities to assess the adequacy of the product

\(^3\)employees in charge of the main architecture of the software
security by pointing out security flaws in the design but most think that something has to happen, like a security breach, to get the customers interested in security. The ISVs think that today customers take security for granted and if something goes wrong the reputation will be damaged and it will be much more expensive to fix in retrospect. The question is too often about money and not about the benefits of security. Bad security gives bad reputation but good security doesn’t give a good reputation is one ISVs reflection.

Some ISVs think that if you want to introduce a model, method or tool you should go through the development team and that management doesn’t care about things like that. Others think that the information should go out to both developers and management and that the developers should have some kind of a workshop or other hands on approach to assimilate the best. They also think it’s important to introduce security in the university educations; "if that doesn’t change the business wont either". The key for better knowledge and awareness, they agree on, is to push on training.

The conclusion is that for a change to take place, something must happen (a security breach) and a general discussion about security must be raised. Until now, no big flaws have been discovered and no known trespasses has been made at the ISVs we’ve been in contact with. One company had a customer that made a audit on their product and found some flaws but they were mostly due to configuration. Because of the ongoing transition to agile methods one company has raised the question about security.

### 5.3.3 The Future

The ISVs think that it’s important to get security in the development process from the start, especially in the design phase and implementation phase. Most of the ISVs want to be able to put aside resources for threat modeling, but what almost all agree on is that customers need knowledge of security and needs to start demanding more secure products.

Even though security isn’t at the front in the software industry today, most of the ISVs we have been talking to think it will be in the future. A rise in demands from customers has been noticed and they think it will get even higher soon. The start of cloud techniques and services\(^4\) is the new thing everyone is talking about and with it the need for privacy and security will be immediate.

### 5.3.4 The SDL

Most of the ISVs know nothing or very little about the SDL or any other security process but all of them seemed interested and some of them can even see the

\(^4\)Also known as Cloud computing, Internet- ('cloud-') based development and use of computer technology ('computing').
company introducing it in their development process in the future. A big key factor here would be if there will be available tools and templates that integrate the method automatically in their development so they don’t have to think that much about it. It must also fit their existing development method which is agile.

Some check-in rules and policies exist today but they are not there because of security. Code analysis is also used to some extent but is adapted to their needs.
Chapter 6

Analysis & Discussion

Here we combine the knowledge we have accumulated in the theory chapters with the information we have gathered in our empirical research. Based on what we conclude from this we discuss and add our own view of the matter.
6.1 Why Security in the Development Processes?

If we look back to chapter 3 where we talked about the essence of computer security, which is to protect the so called CIA-criteria\(^1\). A lot of reasons can be listed for implementing security in software. Protecting privacy, secrecy and trust are the three main arguments that Microsoft gives. We agree with the mainstream of the security community but we also want to emphasize that security not only affects the practical parts of the CIA-criteria but also reputation, trust and profits. In this thesis so far we have presented two ways of achieving secure software. First, by patching when something goes wrong and rely on outside security like encryption, firewalls, access control etc. Second, building in security from the ground up. To do this a SDLC is needed. You don’t necessarily need to get more security out of using a SDLC, even though many studies points towards that you do (see chapter 4.5). The main reason we recommend the second alternative have been mentioned several times before; time and money. The funny thing is that many times this is the main reason why companies don’t want to put resources into security. Why is it that security experts together with studies (see chapter 4) say that a security process will save time and money but ISVs thinks the opposite? We think that the answer is that mid-sized ISVs makes plans for the present and can only see the short term expenses, while security is a long term investment. Even if security is not a priority today we believe it needs to be included in software development in the near future. Especially with the industry moving towards cloud techniques since it is demanded by the customers. More or less all software produced today is connected or going to be connected to the internet. With more connectivity and availability comes more threats and the need for security rises.

6.2 Situation Analysis

The rows we have highlighted in the table (figure 6.1) are the most remarkable findings we have made in the situation analysis. We had a feeling that the maturity level was going to be low, but not this low. It’s really bad and the ISVs need a big push in the right direction to get ready for the upcoming cloud services. The second noticeable finding is the low effort the companies put in educating their co-workers in security. This can of course be connected to the third highlighted row, estimated vulnerability level, where the ISVs are the only ones who think that their vulnerability level is low. The remarkable thing is that they have a security awareness as you can see in the figure, and are at least "medium-aware", but that doesn’t affect their attitude towards education in the area. At least the ISVs exceeded the expectations about using a development process, almost all of them used an agile method. This is a good starting point and if the maturity level, as we hope, will rise there is something to build on. As we described in chapter 4.3, there is a SDL-Agile method that would be a great and suiting thing to start with at the mid-sized ISVs that are using an agile metod.

\(^1\)Confidentiality, integrity and availability.
6.3 The Relevance of the Microsoft SDL

The Microsoft SDL is one of the first and most extensive development lifecycles for security and has therefore received a lot of attention. We think Microsoft have made an important effort to spread the word (chapter 4.6) about the benefits that comes with using the SDL and here in we can see a really big commitment which is outstanding. In spite of this it’s important to remember that even though the SDL is recognized as a really good method, there are others that can be combined to reach the covet result. All of the valid statistics we could find on the outcomes of using the SDL (chapter 4.5) has been unanimous and pointed at large improvements in the software from a security and vulnerability point of view. Microsoft has urged others to do the analysis themselves if they don’t believe their statistics. Inspite this we have not found any reports that have contradicted Microsofts results. There is no silver bullet for security and you might want to adapt the SDL some to your conditions, as the experts recommended (chapter 5.2), and pick the raisins out of the cake. Microsoft themselves are open about this and says that the method will never be perfect or complete but will need ongoing work to be relevant in the future. They also continually publish white papers, templates and guidelines to help others that want to adopt the model. The basic SDL process was built for development of "big" products like OSs and can be a bit heavy for smaller projects. The SDL-Agile (chapter 4.3) can be of help here and be extra helpful for our target group. It’s a good first step to take, but some adaption is still needed for your organization. The experts agree on that the SDL Pro Network is a really good initiative (chapter 5.2) that could be of great help to others, which is something we adhere to, and hopefully this staking will be expanded in the near future.

There is also management friendly proof to the success of the SDL and this in-
formation is important to push on and complement with more data for decision-making that targets mid-sized ISVs. Management approval is, as Microsoft points out, very important for the success of the SDL (chapter 4.4) and must be remembered in the marketing of the process.

One of the changes regarding the SDL that should be made is first and foremost that more resources should be put into promoting the SDL in Sweden, this has to be prioritized. Adding things like abuse cases, requirement formulation, verification and vulnerability scanning tools for reporting statistics is also something we agree with the experts that should be done. Run-time testing tools should also be put under consideration. We got hung up on the fact that the SDL doesn’t consider security requirement elicitation at all. Microsoft uses the SDL as their security requirements in all projects and therefore don’t need to elicit new ones every time. But this will not work for smaller companies. They need explicit requirements to follow, preferably from customers, to consider putting resources into security. We think it’s essential that Microsoft keep putting helpful information out there, one tip is to keep trying to learn about the ISVs perspective and put information out according to their needs. Hopefully this paper will be of some help in this matter. We’ve learned that there is a need for more specified information that different target groups can relate to, like specific cases regarding mid-sized ISVs. This combined with more media attention will help in getting our target group to adopt the SDL. Guidelines, templates and tools that are integrated into the development and project tools are also very important to our target group.

6.4 Where and How can Changes be Made?

So far we have concluded that security is needed and the SDL process can provide it. How will we get this out there? Information, information, information! Media have a strong influence in the software industry but we can’t afford to sit idly by and wait for something to happen to wake the community up. We strongly believe that the only way to get mid-sized software developers to invest in security is through the spreading knowledge. The important thing to work on here is the way the information is presented and how it’s spread. First we want to argue that the concept of security is way too big for non-security experts. It’s impossible to get a deep understanding in security when you are caught up in your own field. That being said, as much as possible should be automated in development tools like Visual Studio. The ISVs also point out that if security activities are integrated in development tools they are more likely to use them than if the tools come separate. Further more the security area needs to be split up and have several different "genres". When you seek out information about security you can find a lot, but far from all of it is useful for your specific needs. We suggest presenting the concept of security as a tree structure where you can go deeper and deeper down until you find what’s eligible for your products. A security model needs to be simple and preferably constructed as a checklist with activities. Even if all the above is achieved some sort of deeper understanding of security is needed when
you adopt a security process. The best way is to hire security experts to guide your company through the transition. A good start, and what we recommend for ISVs, is to either hire someone for the role or make a co-worker the company’s security advisor.

Who do you need to convince to get the ball rolling? The decision to change a organization’s development process will finally be up to the management. We agree that it’s important to get management on the "security train" but we also want to highlight the following; if you successfully convince management, developers are forced to follow. If you successfully convince developers; the management are likely to follow. Management are the first ones that need to be convinced, but if the developers don’t support the changes, mistakes will be made and the security process will not give the desired effect. Another idea is to introduce security process certifications where organizations can certify themselves for a security process, for example for the SDL. This should be similar to how other development process certifications works, like scrum certifications.
Chapter 7

Results

Here we go through each question at issue and present our results, we also suggest some further work that could be performed on the subject.
7.1 How is the term security defined in the context of secure development processes?

Secure development processes involves the whole development lifecycle, everything from preparatory security education, non-functional security requirement, design principles, security to privacy testing to having a response plan when something happens. Security in this context is what’s built in in your product and not the surrounding infrastructure protections. Security should exist in the product before the code is implemented.

7.2 How does the SDL make products more secure?

The SDL process has several requirements of activities you have to do in each phase in the development process. These activities are what makes the development process more secure. One activity doesn’t give more security than another, all of them are important ("A chain is only as strong as the weakest link"). Below is a listing of the activities in the SDL that makes the products more secure.

- All developers, testers and program managers must complete at least one security training class each year.
- At least 80 percent of the project team staff who work on products or services must be in compliance with the concepts listed as "Basic security concepts" in figure 4.3 before their product or service is released.
- Assigning a security advisor and building a security leadership team.
- Determining a bug bar and making sure that the bug tracking system includes security and privacy bug fields.
- Performing a security risk assessment and determining a privacy impact rating.
- Identifying and following design principles.
- Reducing the attack surface as much as possible and performing threat modeling.
- Creating support documentation and tools that address security for users.
- Identifying and following coding best practices.
- Performing security and privacy testing, a security push and a FSR.
- Creating and following a response plan which is updated when needed.
7.3 What is the security experts view of secure development amongst mid-sized ISVs today?

The answer to the question above is presented in more detail in chapter 5. Here we can see a summary of how the experts opinions are on the secure development amongst mid-sized ISVs.

<table>
<thead>
<tr>
<th>Existing process</th>
<th>MEDIUM</th>
<th>Information flow</th>
<th>LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing security process</td>
<td>LOW</td>
<td>Education</td>
<td>LOW</td>
</tr>
<tr>
<td>Security awareness</td>
<td>LOW</td>
<td>Feedback</td>
<td>LOW</td>
</tr>
<tr>
<td>Maturity</td>
<td>LOW</td>
<td>SDL awareness*</td>
<td>HIGH</td>
</tr>
<tr>
<td>Legislation</td>
<td>MEDIUM</td>
<td>Estimated vulnerability level</td>
<td>HIGH</td>
</tr>
<tr>
<td>Future security awareness</td>
<td>MEDIUM</td>
<td>How important is security?</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Knowledge</td>
<td>LOW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 7.1.** Results from our interviews with the security experts. *How aware the experts are of the SDL*

7.4 What changes would mid-sized ISVs benefit from?

Pretty much any of the activities mentioned in 7.2 would benefit mid-sized ISVs in making more secure products. As we mentioned earlier all of the activities in the SDL is needed to produce secure products but a change will not happen over night. What we recommend is to start off with assigning a security advisor at the company that can push for security and start educating the employees. Start where vulnerability is most pressing and add more activities over timer. With the knowledge we have gained through our empirical study we would recommend ISVs to start implementing the high-ranked activities (see figure 7.2) first and work their way down to the low ones.

The majority of our target group use agile methods and the SDL-Agile process will definitely be of use. We believe that, because of the popularity of agile methods, the way to spread information and strengthen the appeal of the SDL is by informing about SDL-Agile. A security activity which we believe missing in the SDL and is needed by ISVs is security requirement elicitation.
<table>
<thead>
<tr>
<th>Activity</th>
<th>Priority</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security training</td>
<td>HIGH</td>
<td>Reduce attack surface</td>
</tr>
<tr>
<td>Verify that the project is eligible SDL</td>
<td>LOW</td>
<td>Threat modeling</td>
</tr>
<tr>
<td>Assign security advisor</td>
<td>MEDIUM</td>
<td>Create support documents</td>
</tr>
<tr>
<td>Assign security leadership team</td>
<td>LOW</td>
<td>Follow coding best practices</td>
</tr>
<tr>
<td>Determine bug bar</td>
<td>MEDIUM</td>
<td>Security and privacy testing</td>
</tr>
<tr>
<td>Security risk assessment</td>
<td>MEDIUM</td>
<td>Security push</td>
</tr>
<tr>
<td>Privacy impact rating</td>
<td>LOW</td>
<td>FSR</td>
</tr>
<tr>
<td>Identify and follow design principles</td>
<td>HIGH</td>
<td>Create a response plan</td>
</tr>
<tr>
<td>Assign security leadership team</td>
<td>LOW</td>
<td></td>
</tr>
<tr>
<td>Determine bug bar</td>
<td>MEDIUM</td>
<td></td>
</tr>
<tr>
<td>Security risk assessment</td>
<td>MEDIUM</td>
<td></td>
</tr>
<tr>
<td>Privacy impact rating</td>
<td>LOW</td>
<td></td>
</tr>
<tr>
<td>Identify and follow design principles</td>
<td>HIGH</td>
<td></td>
</tr>
<tr>
<td>Assign security advisor</td>
<td>MEDIUM</td>
<td></td>
</tr>
<tr>
<td>Determine bug bar</td>
<td>MEDIUM</td>
<td></td>
</tr>
<tr>
<td>Security risk assessment</td>
<td>MEDIUM</td>
<td></td>
</tr>
<tr>
<td>Privacy impact rating</td>
<td>LOW</td>
<td></td>
</tr>
<tr>
<td>Identify and follow design principles</td>
<td>HIGH</td>
<td></td>
</tr>
<tr>
<td>Assign security leadership team</td>
<td>LOW</td>
<td></td>
</tr>
<tr>
<td>Determine bug bar</td>
<td>MEDIUM</td>
<td></td>
</tr>
<tr>
<td>Security risk assessment</td>
<td>MEDIUM</td>
<td></td>
</tr>
<tr>
<td>Privacy impact rating</td>
<td>LOW</td>
<td></td>
</tr>
<tr>
<td>Identify and follow design principles</td>
<td>HIGH</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7.2. The SDL activities prioritized according to order to introduce at mid-sized ISVs

7.5 Further work

One thing that we thought was interesting, that came up during our survey, was the classification of security activities. Unfortunately we did not have the time go deeper into this during our studies. We think it could be a good subject for further work and there definitely exists a need for it amongst mid-sized ISVs. It could entail research about how to split up the business in different genres and associate them with necessary security activities. As we mentioned before in chapter 6 we suggested to present the information as a tree structure that is easy to follow to find appropriate security measures.
Bibliography


Appendix A

SDL-Agile Requirements

In this appendix we list all the requirements for SDL-Agile. [20]
<table>
<thead>
<tr>
<th>Title</th>
<th>Requirement/Recommendation</th>
<th>Applies to Online Services</th>
<th>Applies to Managed Code</th>
<th>Applies to Native Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communicate privacy-impacting design changes to the team’s privacy advisor</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Compile all code with the /GS compiler option</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Comply with SDL firewall requirements</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do not use banned APIs in new code</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ensure all ASP.NET applications use the ValidateRequest cross-site scripting input validation attribute</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Ensure all database access is performed through parameterized queries to stored procedures</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ensure all team members have had security education within the past year</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ensure the application domain group is granted only execute permissions on the database stored procedures</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Fix all issues identified by code analysis tools for unmanaged code</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fix all security issues identified by CAT.NET and FxCop static analysis</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Follow input validation and output encoding guidelines to defend against cross-site scripting attacks</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Link all code with the /dynamicbase linker option (Address Space Layout Randomization)</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link all code with the /Xcompat linker option (Data Execution Prevention)</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link all code with the /defines linker option (safe exception handling)</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update threat models for new features</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Use HeapSetInformation</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use the /robust MIDL compiler switch</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use the Relying Party Suite SDK</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Avoid JavaScript eval function and equivalents</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Canonicalize URLs</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Encode long-lived pointers</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Review error messages to ensure sensitive information is not disclosed</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Use standard annotation language (SAL) to annotate all functions</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use strict /GS option</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use whitelist of allowed domains to perform redirects</td>
<td>Requirement</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Title</td>
<td>Bucket A: Security Verification</td>
<td>Requirement/Recommendation</td>
<td>Applies to Online Services</td>
<td>Applies to Managed Code</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------</td>
<td>-----------------------------</td>
<td>----------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Debug the application with the Application Verifier enabled</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Disable tracing and debugging in ASP.NET applications</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Investigate and service any reported / GS crashes</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform ActiveX control fuzzing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform attack surface analysis</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform binary analysis (BinScope)</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform CDM object testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform cross-domain scripting testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform file fuzz testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Perform RPC fuzz testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Conduct in-depth manual and automated code review for high-risk code</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform data flow testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform input validation testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Perform replay testing</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Bucket B: Design Review</th>
<th>Requirement/Recommendation</th>
<th>Applies to Online Services</th>
<th>Applies to Managed Code</th>
<th>Applies to Native Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoid cross-domain access to authenticated sites</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comply with User Account Control (UAC) best practices to ensure all code runs as a non-administrator</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conduct a privacy review</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensure all code is compliant with the SDL Cryptographic Standards</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensure all code is compliant with the SDL Privacy Guidelines document</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use strongly named assemblies, and request minimal permissions</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete in-depth threat model training</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Disable rarely used features by default, to reduce attack surface</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Grant minimal privileges</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Review planning and design specifications for user interface elements</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Use Windows Imaging Component to process image data</td>
<td>Requirement</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Title</td>
<td>Requirement/Recommendation</td>
<td>Applies to Online Services</td>
<td>Applies to Managed Code</td>
<td>Applies to Native Code</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td>----------------------------</td>
<td>-------------------------</td>
<td>------------------------</td>
<td></td>
</tr>
<tr>
<td>Add or update privacy scenarios in the test plan</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create or update the list of response contacts</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Define or update the privacy bug bar</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Define or update the security bug bar</td>
<td>Requirement/Recommendation</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Ensure symbols are available internally for all public releases</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create or update a business continuity-disaster recovery plan</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create or update a network down plan</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Create or update content publishing plan</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Create or update privacy support documents</td>
<td>Requirement</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>Requirement/Recommendation</td>
<td>Completion (months)</td>
<td>Applies to Online Services</td>
<td>Applies to Managed Code</td>
<td>Applies to Native Code</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td>---------------------</td>
<td>-----------------------------</td>
<td>--------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Avoid writable PE segments</td>
<td>Requirement</td>
<td>6</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Configure bug tracking to track the cause and effect of security vulnerabilities</td>
<td>Requirement</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create a baseline threat model</td>
<td>Requirement</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Determine security response standards</td>
<td>Requirement</td>
<td>6</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Establish a security response plan</td>
<td>Requirement</td>
<td>6</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Identify primary security and privacy contacts</td>
<td>Requirement</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Identify your team's privacy expert</td>
<td>Requirement</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Identify your team's security expert</td>
<td>Requirement</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Use approved XML parsers</td>
<td>Requirement</td>
<td>6</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Use latest compiler versions</td>
<td>Requirement</td>
<td>12</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Designate full-time security program manager</td>
<td>Requirement</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Remove dependencies on NTLM authentication</td>
<td>Requirement</td>
<td>12</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Appendix B

Microsoft SDL: Return on Investment, Project Example
A small development organization is considering a new security effort but is unsure about the costs and their capabilities. They have a team of approximately 50 developers working on about five projects a year. Management is very interested in security, in theory, but they are resource-strapped, in practice. Being able to fulfill multiple requirements with the same activities is a priority. Following the recommendations described earlier, they build a fairly small and very front-loaded process to take the maximum possible advantage of structural efficiencies in the absence of a legacy process. They start with a single pilot project for a full security review, and they bring in outside experts to help. In addition, they start to do high-level review activities on two more projects. As data comes in from these initial activities, they expand to a larger set of activities. In addition to looking at the security benefits, metrics are chosen to examine the overall reliability and quality of the systems to help justify the new security expenditure in light of overall efficiency.

**Enterprise-scope security activities:**

<table>
<thead>
<tr>
<th>Activity</th>
<th>Cost</th>
<th>Metrics used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensive training for five architects/year</td>
<td>$25k</td>
<td>Requirements and architecture-influenced vulnerability delta</td>
</tr>
<tr>
<td>Standards and policy training for 50 developers/year</td>
<td>$10k</td>
<td>Standards compliance delta</td>
</tr>
<tr>
<td>Standards and policy creation and updating</td>
<td>$5k</td>
<td>Vulnerability delta for covered issue types</td>
</tr>
<tr>
<td>Incident response planning</td>
<td>$3k</td>
<td>Incident cost delta versus industry, response time</td>
</tr>
<tr>
<td>Communication and process planning</td>
<td>$3k</td>
<td>Cost delta for equivalent security activities, non-security development maturity</td>
</tr>
<tr>
<td>Enterprise-scope metrics tracking</td>
<td>$2k</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$48k</strong></td>
<td></td>
</tr>
</tbody>
</table>
Initial project-scope security activities:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Cost</th>
<th>Metrics used</th>
</tr>
</thead>
<tbody>
<tr>
<td>External full-access security review for one project</td>
<td>$80k</td>
<td>Severity-normalized vulnerabilities found per person-week</td>
</tr>
<tr>
<td>Requirements review for three projects</td>
<td>$16k</td>
<td>Requirements-level security vulnerabilities, business-process violation cost estimate, non-security project quality improvements</td>
</tr>
<tr>
<td>Security analysis/threat modeling for three projects</td>
<td>$15k</td>
<td>Architecture-level security vulnerabilities, non-security project quality improvements</td>
</tr>
<tr>
<td>Security specification creation for three projects</td>
<td>$8k</td>
<td>Vulnerability delta for covered issue, types versus similar applications</td>
</tr>
<tr>
<td>Security Vulnerability Tracking</td>
<td>$2k</td>
<td>N/A</td>
</tr>
<tr>
<td>Project metrics tracking</td>
<td>$2k</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$123k</strong></td>
<td></td>
</tr>
</tbody>
</table>

Eventual Project-Scope Security Activities:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Cost</th>
<th>Metrics used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal full access security review for two projects/year</td>
<td>$120k</td>
<td>Severity-normalized vulnerabilities found per person-week</td>
</tr>
<tr>
<td>Requirements review for five projects/year</td>
<td>$20k</td>
<td>Requirements-level Security vulnerabilities, business-process violation cost estimate, non-security project quality improvements</td>
</tr>
<tr>
<td>Security analysis/threat modeling for five projects/year</td>
<td>$30k</td>
<td>Architecture-level security vulnerabilities, non-security project quality improvements</td>
</tr>
<tr>
<td>Security specification creation for five projects/year</td>
<td>$10k</td>
<td>Vulnerability delta for covered issue, types versus similar applications</td>
</tr>
<tr>
<td>Security vulnerability tracking</td>
<td>$2k</td>
<td>N/A</td>
</tr>
<tr>
<td>Project metrics tracking</td>
<td>$2k</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$184k</strong></td>
<td></td>
</tr>
</tbody>
</table>

Half the battle is knowing how large the investment is and knowing the metrics we are tracking. Now we need to look at how these activities improve our resource utilization.

Cost estimates for vulnerabilities classes
These cost estimates are based on the cost of fixing a vulnerability based on the phase in which it is discovered. Although using real data is optimal, any metrics on vulnerability costs can be used to calculate this for pre-release vulnerabilities; they do not necessarily have to be for security vulnerabilities. Additional costs are added for high- and medium-severity vulnerabilities discovered in the deployment phase to account for actual incident losses or downtime to patch.
Number of vulnerabilities discovered or prevented during initial activities:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Vulnerabilities discovered</th>
<th>Savings versus discovery in production</th>
</tr>
</thead>
<tbody>
<tr>
<td>External full access security review</td>
<td>2 high</td>
<td>($80k x 2) - ($15k x 2) +</td>
</tr>
<tr>
<td>for one project</td>
<td>3 medium</td>
<td>($40k x 3) - ($15k x 3) +</td>
</tr>
<tr>
<td></td>
<td>10 low</td>
<td>($30k x 10) - ($15k x 10) =</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Subtotal: $355k</td>
</tr>
<tr>
<td>Requirements review for three projects</td>
<td>1 high</td>
<td>($80k x 1) - ($1k x 1) = $79k</td>
</tr>
<tr>
<td></td>
<td>1 medium</td>
<td>($40k x 1) - ($1k x 1) = $39k</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Total savings: $437k</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cost of work: $123k</td>
</tr>
</tbody>
</table>
|                                      | Total                       | 3 high, 4 medium, 10 low               | $473k - $123k = $350k ROI

This clearly shows the return received on our investment, since the cost to fix these issues before production is much lower than the combined cost of fixing them earlier in the process and the cost of the security activities throughout the lifecycle.
Appendix C

Interview Questions
Questions to security experts:

1 Vad utvecklas till största del här hos er?
2 Utvecklar ni någon egen produkt?
3 Hur stort är ett "vanligt" projekt med tanke på tid, antal personer på utvecklingsteamet etc?
4 Vad har företaget för policy för säkerhet?
5 Är det några lagar som berör er när det kommer till säkerhet?
6 Tror du säkerhet i er utveckling påverkar företagets rykte?
7 Hur används säkerhet när ni utvecklar? Intergeraras det i utvecklingen? Används en modell? Finns möjlighet till information omkring den vi kan studera i efterhand?
8 I vilken fas i projektet börjar ni införa säkerhet?
9 Ställs särskilda säkerhetskrav i kravhanteringsfasen fasen? Hur prioriteras dom?
10 Finns det någon säkerhetsansvarig på varje projekt? Vem/vilka?
11 Är det nånting i processen som du saknar?
12 Utbildas era anställda när det kommer till säkerhet? Om JA: Regelbundet?
13 Leder modell till några synliga förändringar i merarbete efter leverans?
14 Vilken är er viktigaste säkerhetsåtgärd? Varför?
15 Använder ni threath modeling? 16 När konsulter hos er är anlitade till ett icke säkerhetsrelaterat projekt, har dom någon inflytande i säkerheten i utvecklingsprocessen?
17 Har någon mjukvara ni producerat råkat ut för en säkerhets-(incl privacy)incident?
18 Påverkar säkerhetsarbetet kundnöjdheten? Hur då?
19 Finns det fall då säkerhet extra viktigt? vilka?
20 Tar man till vara på kunskapen efter avslutade projekt? Hurdå?
21 Varifrån får man uppgifter om nya säkerhetsproblem?
22 Används några kodanalysprogram?
23 Hur testas/verifieras det ni utvecklar? Kollar ni efter vad som kan gå fel eller kollar ni så det fungerar?
24 Har ni en plan för åtgärder om en sårbarhet upptäcks efter leverans?
25 Vem står för kostnaderna om mjukvaran ni levererat måste patchas?
26 På en skala 1-5, hur säkra skulle du ranka det ni utvecklar?
27 Hur är inställningen för säkerhet hos kunder? (projektledare, utvecklare, tekniskchef...)
Allmänt säkerhet

28 På en skala 1-5, hur viktigt tycker du säkerhet är i mjukvara?

29 Vad tror du skulle behövas för att få ert företag att tänka mer på säkerhet?

30 Om man skulle visa på papper att implementera en viss typ av säkerhetsprocess skulle bidra till X minskning i utgifter, tror du att det skulle hjälpa med att få in säkerhet i er utveckling?

31 Vad är din inställning till säkerhet? Ett sätt att plugga igen säkerhetsläckor efter hand med befintliga verktyg och features, eller bygga in säkerhet från grunden?

32 Vem/vad påverkar hur mycket säkerhetstänk som finns inom företag?

33 Tror du ert säkerhetsarbete kommer utvecklas i framtiden?

SDL

34 Vad vet du om SDL? Är det något du tror ert företag skulle kunna tänkas använda?

35 Vad tycker du om SDL?

36 Har du någon egen erfarenhet/kommit i kontakt med/av SDL?

Svar JA:

37 Hurdå?

38 Vad för typ av projekt användes det i?

39 Hur tyckte du det fungera?

40 Gav det väntat resultat?

41 Några nackdelar du såg i det arbetet som var SDL relaterat?

42 Slavade ni efter SDL-modellen, eller användes bara specifika bitar?

43 Tycker du att hela SDL måste användas eller kan man bara använda vissa bitar?

44 Den mesta information som vi har kommit över har varit mycket SDL glorie rad. Tycker du det finns några negativa sidor hos SDL i nuläget? Vad tycker du om SDL?

45 Känner du till SDL Pro Network?

Svar JA:

46 Vad tycker du om det initiativet från MS?

47 Har du hört nått om hur samarbetet går, nu när det nästan gått ett år sen start?

48 Vet du om det är planer att SDL Pro Network ska börja utvecklas i Sverige?
Svar NEJ (Berätta ...):

49 Vad tycker du om det initiativet från MS?

50 Tror du det är något som skulle välkomnas här i Sverige?

51 Känner du till någon annan säkerhetsmodell?

52 Känner era kunder till SDL?

Questions to ISVs:

1 Vad är din arbetsroll?

2 Vad för typ av produkter utvecklas i största dels här hos er?

3 Hur stora är ett "vanligt" projekt hos er med tanke på tid, antal personer på utvecklingsteam etc?

4 Använder ni någon typ av utvecklingsmodell eller process när ni arbetar med era produkter?
   Om JA:
   5 Vad använder ni?
   6 Följs den blint?
   7 Används nån typ av threat modeling?
   Om NEJ:
   8 Används nån informell utvecklings modell? Hur används den? Grovt vilka steg ingår och hur fungerar den (Ingår krav, design, implementation, test?)?
   9 På en skala 1-5, hur viktigt tycker du säkerhet är i mjukvara?
   10 Har företaget någon policy för säkerhet?
   11 Är det några lagar som berör er när det kommer till säkerhet?
   12 Tror du säkerhet i era produkter påverkar företagets rykte?
   13 Tänker ni på säkerhet när ni utvecklar era produkter?
   Om JA:
   14 Hur då? Intergeraras det i utvecklingen? Används en modell? Finns möjlighet till information omkring den vi kan studera i efterhand?
   15 I vilkenfas i projektet börjar ni införa säkerhet?
   16 Ställs särskilda säkerhetskrav i kravhanteringsfasen fasen? Hur prioriteras dom?
   17 Finns det någon säkerhetsansvarig på varje projekt? Vem/vilka?
   18 Är det nånting i processen som du saknar?
   19 Utbildas era anställda när det kommer till säkerhet? Om JA: Regelbundet?
   20 Budgeteter ni för säkerhet? Hurdå?
   21 Sparar ni eller förlorar ni pengar på säkerhet?
   22 Leder modellen till några synliga förändringar i merarbete efter leverans?
   23 Vilken är er viktigaste säkerhetsåtgärd? Varför?
   24 Vilken är den mest effektiva säkerhetsåtgärden?
Om NEJ:
25 Tycker du ni bör/behöver tänka på säkerhet när ni utvecklar era produkter? Om JA:
26 Vad är det som hindrar er? (bromsande krafter?)
27 Vad tror du skulle behövas för att få ert företag att tänka mer på säkerhet?
28 Om man skulle visa på papper att implementera en viss typ av säkerhetsprocess skulle bidra till X minskning i utgifter, tror du att det skulle hjälpa med att få in säkerhet i er utveckling?
29 Vad är din inställning till säkerhet? Ett sätt att plugga igen säkerhetsläckor efter hand med befintliga verktyg och features, eller bygga in säkerhet från grunden?
30 Vad använder ni för utvecklingsverktyg/plattform?
31 De flesta verktyg har inbyggda säkerhetsåtgärder använder ni dem? Tycker du det saknas nått i de verktygen?
32 Använder ni några kodanalysprogram?
33 Hur testas/verifieras era produkter? Kollar ni efter vad som kan gå fel eller kollar ni så det fungerar?
34 Hittar ni/rapporterar det fel/säkerhetsbrister i era produkter i efterhand?
35 Har ni en plan för åtgärder om en sårbarhet upptäcks efter leverans?
36 Har någon mjukvara ni producerat råkat ut för en säkerhets-(inkl privacy)incident?
37 Vem står för kostnaderna om mjukvaran ni levererat måste patchas?
38 På en skala 1-5, hur säkra skulle du ranka era produkter?
39 Finns efterfrågan på säkerhet från era kunder?
Om JA
40 Tillhandahåller ni det då?
41 År det vanligt att kunderna ställer säkerhetskrav?
Om NEJ
42 Varför inte tror du?
43 Påverkar säkerhetsarbetet kundnöjdheten? Hur då?
44 Finns det fall då säkerhet extra viktigt? Vilka?
45 Vilka aktiviteter brukar ni använda er av när extra säkerhet krävs?
46 Har ni nån som är ansvarig för säkerhet på företaget?
Om JA
47 Vem och vad är den personens ansvar?
   Om NEJ

48 Skulle det finnas intresse av att ha en sådan person?
   Om JA

49 Vad är hindret?

50 Hur är inställningen för säkerhet hos medarbetarna? (projektledare, utvecklare, teknisk chef...)

51 Tar man till vara på kunskapen efter avslutade projekt? Hurdå?

52 Har de anställda kunskaper inom säkerhet? Varifrån?

53 Känner du till SDL (Security Development Lifecycle)?
   Om JA

54 Vad vet du om den? Är det något du tror ert företag skulle kunna tänkas använda?
   Om NEJ

   (Berätta ...):

55 Är det något som du tror ert företag skulle vara intresserad av att veta mer om?
   Om JA


57 Känner du till någon annan säkerhetsmodell?

58 Vad tror du behövs för att företag ska börja satsa mer på säkerhet i utvecklingsprocesser?
Questions to Microsoft:

1 Vad gör du på Microsoft?
2 Kommer du i kontakt med säker utveckling? Hur då? Var då? När då?

Säker utveckling

3 Finns ett behov av säker utveckling i mjukvarubranschen?
4 Finns tillräckligt med kunskap om säkerhetsutveckling i mjukvarubranschen? Inom Microsoft Sverige?
5 Hur tillgodogörs kunskapen internt/externt (på andra företag)? Var? Kurser? Vilka får delta?
6 Finns det efterfrågan på metoder för säker utveckling? Var finns efterfrågan?
7 Vad är ditt perspektiv på säker utveckling?
8 Hur ser du på Microsofts roll i mjukvarusäkerhet?
9 Tror du att mjukvaruföretag utanför Microsoft är redo att satsa på säker utveckling?

SDL

10 Vad har du för spontana tankar kring SDL? (vad är det första du tänker på)
11 Har du någon egen erfarenhet/kommit i kontakt med/av SDL?
Svar JA:
12 Hurdå?
13 Vad för typ av projekt användes det i?
14 Hur tyckte du det fungera?
15 Gav det väntat resultat?
16 Några nackdelar du såg i det arbetet som var SDL relaterat?
17 Slavade ni efter SDL-modellen, eller användes bara specifika bitar?
18 Hur hög är kunskapen om SDL inom Microsoft i Sverige? Intresset?
19 Var hämtas (ny) information?
20 Tycker du att hela SDL måste användas eller kan man bara använda vissa bitar?
21 Tycker du det finns några andra "overall" säkerhets metoder som kan jämföras med SDL?
22 Vilka säkerhetsutvecklings metoder tror du används mest i Sverige idag?
23 Den mesta information som vi har kommit över har varit mycket SDL glorifierad. Tycker du det finns några negativa sidor hos SDL i nuläget?
24 Jag läste på Bryan Sullivans blogg där han prata lite om säkerhet i webbutveckling och SDL. Han nämnde att dom flesta webbapplikationer inte tar ett par år att utveckla som större projekt gör, utan istället ett par veckor. Han menar att den korta utvecklingstiden är ett problem för SDL. Stämmer det? Varför? Existerar gränser?

25 Något du personligen saknar i SDL-processen?

26 Hur sprids information om SDL idag?

27 När du/ni är ute och föreläser, nämns SDL i din/era presentationer? Ställs det frågor kopplat till SDL?

28 Vem har best koll på SDL i Sverige?

SDL Pro Network

30 Känner du till SDL Pro Network?

Svar JA:

31 Vad tycker du om det initiativet från MS?

32 Har du hört nått om hur samarbetet går, nu när det nästan gått ett år sen start?

33 Vet du om det är planer att SDL Pro Network ska börja utvecklas i Sverige?

Svar NEJ (Berätta ...):

34 Vad tycker du om det initiativet från MS?

35 Tror du det är något som skulle välkomnas här i Sverige?

SDL Optimization Model

36 SDL Optimization Model existerar för att komma igång med SDL, varför ska man gå med i SDL Pro Network?

SDL-verktyg

(SDL Threat Modeling VSTS template TFS .NET Framework)

37 SDL är integrerat i VSTS och TFS, används det mycket utanför Microsoft?


Microsoft Partners

39 Hur mycket tror du SDL används utanför Microsoft?

40 Tror du det finns behov för SDL hos MS-partners och i övriga mjukvarubranschen?

41 Är det stor efterfrågan på säkerhet bland de MS-partners du har kontakt med?

43 Har MS planerade resurser för att aktivt erbjuda kunskap om SDL till partners? Utbildningsunderlag?

44 Vad för typ av hjälp erbjuder Microsoft deras partners som är villiga att adoptera SDL?


46 Vilka MS-partners tror du är bra för oss att besöka?

**Framtid**

47 Hur tror du det kommer att se ut i framtiden med mjukvaruutveckling (2/5/10 år)?

48 Hur ser framtiden ut för SDL i Sverige? Vill du bidra till den?

49 Finns det nått intresse hos Microsoft att engagera sig i skolutbildningen, angående säkerhet, för framtidens utvecklare?

**Vårt bidrag**

50 Vad skulle du vilja se att en undersökning om SDL hos MS-partners ska ge?

51 Vad tror du att en sådan undersökning kommer ge?

52 Om vår undersökning hos MS-partners skulle visa att största delen av all utveckling inte inkluderar säkerhet alls, hur skulle det påverka ert arbete framöver?

53 Med samma utfall som ovan, hur tror du man bäst övertygar MS-partners att tänka mer på säkerhet?

**Övrigt** 54 (Kanske en sidnot) Länder som Kina och Indien växer allt snabbare och större inom mjukvaruutveckling, Hur stor säkerhets "awarness" tror du det finns där?