AGILE SECURITY METHODS: AN EMPIRICAL INVESTIGATION

Ahmed Alnatheer, Andrew M. Gravell, David Argles, Lester Gilbert
ECS, University of Southampton
Southampton, SO17 1BJ, UK
aaa1v09@ecs.soton.ac.uk, amg@ecs.soton.ac.uk, da@ecs.soton.ac.uk, lg3@ecs.soton.ac.uk

ABSTRACT
This paper provides an in-depth investigation into the various security issues in Agile software development methodologies currently in use. We have as part of our research identified a number of issues from multiple perspectives and points of view throughout the literature and from industrial sources. These shed light into what the most important issues are and what is the best way forward in assessing each proposal and deciding whether to adopt it or not. We have conducted a number of empirical interviews with practitioners from various parts of the world who actively work in high risk projects which puts them in a position to shed light into these topics in a more detailed fashion. We present our findings and analysis in the following paper for the topics of combining security and agility. We assess whether changing Agile methods for the sake of security is really necessary, and present recommendations for future work and conclusions.

KEY WORDS

1. Introduction
Security experts often criticize Agile for having a fundamental lack of an inherent security mechanism to produce secure software [1] but fail to acknowledge that their approach is heavily focused on documentation and practices which hinder the Agile process in many ways if applied without modification. The current proponents of extending Agile methodologies have not yet been able to prove why the current Agile processes such as XP, FDD, and Scrum cannot accommodate security practices as part of the already established methods and why an amendment or extension is required. There are a number of solutions proposed by both sides which suggest a set of compromises in order to bring these methodologies closer to integration [2-4]. From among this body of work, a set of ideas, issues, and questions emerged that became part of the motivation of this research to extract, analyze, and provide evidence for. This work is focused primarily on security issues in Agile methods as it is perceived by practitioners working in the field and the conclusions may not be applicable to a wider SDLC discussion on security standards and wider aspects of secure software development.

As part of this effort, we have identified outstanding security issues [5]. We identified a gap in the literature on the lack of empirical works on the topic as well which underscores the importance of having a continued discussion on this important topic which is gaining more widespread attention by the industry in addition to the academic field. In order to gain a better practical understanding of identified issues, we have designed and conducted empirical semi-structured interviews in order to assess how useful and feasible these proposed security solutions could be in practice.

As result of our review of literature and surrounding issues, 68 papers were looked at on the topics of Agile and security, 36 presented issues that were relevant to the topics of security in agile, 17 of which had issues related to integration and/or extension mechanisms but only 6 were empirical which shows the gap in the literature on this important topic. Dingsøyr also described this as urgently needed to be investigated empirically [6].

In this paper, we will outline the results of the interviews along with our conclusions centered around two main research questions:

RQ1. How to seamlessly combine security and agility, and evaluate the suitability of security mechanisms for use in agile?

RQ2. Is changing one or more agile practice(s) for better security really necessary and what kind of projects would require security more than others?

1.1 The interviews
Due to the sensitive nature of the topic being security issues in software development methodologies virtually every credible organization we contacted refused to participate often citing reasons given to them by their legal department on why it couldn’t be done, even though we made assurances of anonymity and confidentiality for both the organization and the participants. This left us to follow through with the next best approach which was to directly interview practitioners outside of their professional work environment in order to gain an insight into their practices and knowledge surrounding the topic.

The subjects were chosen based upon their level of experience and domain knowledge on the issues related to Agile software development and secure software development. The subjects were chosen from a pool of
authors, practitioners, managers, and stakeholders that have related their experience to these topics and are interested to participate in the interview. The selection criteria included being from a background of software engineering and development and related fields that directly impact the agility and security of the final product. Contact was made in person, through the internet, through our supervisors, workshops, and conferences, and after that through various forms of communication such as Phone, Email, etc.

1.2 The candidates
After the ethic approval process was completed and approved, we began our search for suitable candidates and after an extensive survey of the practitioners in the industry both in UK and other countries we were able to secure 15 interviews with 16 suitable candidates. The geographical distribution of the participants was carefully taken into consideration for us to have a representative sample from around the world. Anonymity was provided for participants and they were assured by the researcher that their information was not going to be released in any way to the general public without their express consent due to the extremely sensitive nature of the topics under study and the relative importance of their roles in their respective position in often very reputable organizations.

2. Research Design
The setting for the interviews was partly chosen to be among the professional groups that meet routinely in order to share information and update individuals with the latest tools, training, and technologies. The following table shows the results of the background information for each individual interviewed that lists their professional attributes that could be shared without exposing any personally identifiable information.

2.1 Forms of data collection
As part of semi-structured interviews participants were asked a number of open ended questions. Based upon answers given to those questions, a number of more specific follow-up questions were asked from participants in order to gain a much deeper insight and understanding of the issues under investigation. As part of the questioning, the interviewees were gauged in their manner of response as well as the quality of their answers given

<table>
<thead>
<tr>
<th>No</th>
<th>Exp (yrs)</th>
<th>Location</th>
<th>Interview Length</th>
<th>Method</th>
<th>Iteration Length (weeks)</th>
<th>Iterations per Release</th>
<th>Role</th>
<th>Type of Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>Taiwan</td>
<td>75 min</td>
<td>Scrum + XP</td>
<td>4-5</td>
<td>4-5</td>
<td>Developer</td>
<td>Software/ Systems Development</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Japan</td>
<td>110 min</td>
<td>Other Agile</td>
<td>2-3</td>
<td>3-4</td>
<td>Developer</td>
<td>Electronic Company</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>U.K.</td>
<td>35 min</td>
<td>XP + other Agile</td>
<td>2</td>
<td>1</td>
<td>Developer/QA</td>
<td>Financial organization</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>China</td>
<td>95 min</td>
<td>XP</td>
<td>2-4</td>
<td>3-6</td>
<td>Developer</td>
<td>Software Development Company</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>U.S.</td>
<td>77 min</td>
<td>Scrum</td>
<td>2</td>
<td>24</td>
<td>Developer</td>
<td>Electronic Company</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>U.S.</td>
<td>38 min</td>
<td>XP</td>
<td>per story</td>
<td>Varies</td>
<td>Developer/Manager/QA</td>
<td>Software Developing Company</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>U.K.</td>
<td>30 min</td>
<td>TDD</td>
<td>2</td>
<td>1</td>
<td>Developer/QA</td>
<td>Financial organization</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>U.K.</td>
<td>35 min</td>
<td>Other Agile</td>
<td>2-6</td>
<td>Varies</td>
<td>Consultant</td>
<td>Consultation Firm</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>U.K.</td>
<td>52 min</td>
<td>Other Agile</td>
<td>2</td>
<td>Varies</td>
<td>Security/Software Engineer</td>
<td>Software/ Systems Development</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>U.K.</td>
<td>23 min</td>
<td>Other Agile</td>
<td>Varies</td>
<td>Varies</td>
<td>Manager</td>
<td>Financial organization</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>U.K.</td>
<td>29 min</td>
<td>Scrum</td>
<td>2</td>
<td>8-12</td>
<td>Agile Coach/Manager</td>
<td>Dealer Company</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>U.K.</td>
<td>66 min</td>
<td>Scrum</td>
<td>Varies</td>
<td>Varies</td>
<td>Consultant/Scrum Master</td>
<td>Consultation Firm</td>
</tr>
<tr>
<td>13</td>
<td>28</td>
<td>U.K.</td>
<td>37 min</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Consultant/Manager/Developer</td>
<td>Consultation Firm</td>
</tr>
<tr>
<td>14</td>
<td>21</td>
<td>U.S.</td>
<td>57 min</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Security/ Consultant</td>
<td>Security Firm</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>Saudi Arabia</td>
<td>94 min</td>
<td>Waterfall</td>
<td>N/A</td>
<td>N/A</td>
<td>Manager/ Security Auditor/ Developer</td>
<td>Telecom Company</td>
</tr>
</tbody>
</table>

Table 1: Professional Interviewee Background Information
semi-structured interviews proved to be most useful and appropriate in this situation.

2.2 Rationale for choosing this data collection method
We acknowledge that this method is not as desirable as a direct observation technique since the information that we get is filtered through the views of interviewees but given the sensitive nature of the research that deals with security issues and organization’s lack of cooperation in sharing such information directly we maintain that the data collection method used was the best possible for the research at this time.

2.3 Information gathering protocol
For the semi-structured interviews, we recorded the audio of the majority of the interviews and proceeded to transcribe and summarize the information provided. The steps are as follows:

1. Company information (related to the type of the organization and not the specific name of the company) and their role in the company (their job title and not their name)
2. Started with a general question relevant to the topic of the interview (See Table 2).
3. Based on the answer from the first question, have specific follow-up questions available to ask that may not be necessarily chosen beforehand.
4. Framework for the interview in the form of an interview guide.

Email addresses and phone numbers were used only for the purposes of contacting the individual interviewees.

3. Analysis Steps
A typical qualitative analysis involves the collection of data by the researcher who then proceeds to identify and collect themes or perspectives, and presents them in a concise and human understandable form [7].

We followed a similar approach to Grounded Theory [8-10] with a key difference of using prior research data to generate interview questions for use in the analysis and interpretation stages as well as using data gathered through semi-structured interviews. The type of analysis used follows the Hybrid Approach mentioned in a book titled “Transforming Qualitative Information” by Boyatzis (1998) that outlined steps to follow [11] as part of the analysis and interpretation of qualitative data:

3.1 Preparing data for analysis
There was over 15 hours of raw data captured in the form of audio that was later transcribed into over 140 pages of written textual material. The transcriptions were read over and summarized into 120 pages of relevant information and were subsequently entered into NVIVO by the researcher. From that point the information was analyzed and codified into over 665 general codes which were later reduced to 419 highly relevant and specific instances useful and directly pertaining to the topics of interest. The basis for developing the codes consists of a combination of codes that emerge from the data as well as codes that were pre-determined from the literature.

The 419 codes were then used to identify 34 major themes that were conceptually gathered into 8 groups based on our research questions. For this paper, we will

<table>
<thead>
<tr>
<th>Interview General Questions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What are your project or code level security practices that are in use today in your organization?</td>
<td></td>
</tr>
<tr>
<td>2. Is security as important to agile projects as many would like to have us believe?</td>
<td></td>
</tr>
<tr>
<td>3. Which specific practice in your project contributed the most to increased security of the resulting software?</td>
<td></td>
</tr>
<tr>
<td>4. Which factor(s) are most responsible for causing vulnerabilities and weaknesses in software in terms of security?</td>
<td></td>
</tr>
<tr>
<td>5. Is there an observed or apparent trend towards increased security of software in agile projects?</td>
<td></td>
</tr>
<tr>
<td>6. How can security related aspects such as risk analysis be effectively made to become part of iterative and incremental processes?</td>
<td></td>
</tr>
<tr>
<td>7. Does every project inevitably end up with some security vulnerabilities and what can be done to mitigate this risk?</td>
<td></td>
</tr>
<tr>
<td>8. What is the best way to model Security issues as part of agile projects?</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Sample of Semi-Structured Interview Questions

---

85
only present the data related to a subset of our original research questions.

3.2 Developing themes and codes
As we mentioned earlier most of the themes are emerging (Inductive or Data Driven) while some are based on the literature. Data driven codes were built inductively from the transcribed and later summarized interview data. According to Boyatzis the researcher is responsible for analyzing the data, interpreting the meaning of finding and to build hypotheses after obtaining results. Since the closeness of the code to raw information is high, the likelihood that people will code the raw data similarly is increased and therefore this approach has a relatively higher interrater reliability than other approaches. With a complete view of the information the researcher can recognize “gross (i.e. easily evident) and intricate (difficult-to-discrim) aspects of the information.” [11]. Each theme includes a number of perspectives expressed by individuals that answered a particular question related to the theme and is supported by a number of specific quotes (as evidence). The themes were then organized conceptually based on their relationship to each of our specific research questions and taken collectively formed an interconnected and complex picture.

The unit of analysis is the predominant security issues in Agile teams in which the various stakeholders that were interviewed operate. In contrast, the unit of coding is the smallest part of the raw data that could be evaluated in a useful way regarding the unit of analysis which in our case is the opinions of the individuals that are part of the agile team. These opinions are in the form of a response (or a part of the response) to each interview question.

4. Results of Interviews
According to Coffey and Atkinson (1996), clustering is a way of organizing data to assist in the process of analysis and interpretation [12]. As such, we grouped themes discovered through our semi-structured interviews conceptually based on our research questions and each code discovered would be rated based on the frequency of occurrence once per interviewee per question (consensus index) and presence or absence for measuring popularity of the issues. Similar themes could be applied to more than one research question and we discovered this to be the case during the process of coding and analysis of results. We employed frequency scoring because we wanted to be able to describe and analyze the observation using numeric representation.

4.1 Popularity index
In order to find the significance of each issue among the participants in the interviews, we chose to obtain a frequency score for each participant on the issue(s) they raised during the interview on the various questions that we wanted to discuss that could tell us about their preferences from various perspectives and points of view. The results of the empirical observations gathered from the interview data will help in determining the most predominant and important security issues in Agile methodologies as reported by practitioners.

Our frequency scoring is calculated based on an nominal scale of the number of mentions by the interviewee at various points throughout the interview on various issues discussed. The scale is relative because in order to find which issues are more popular with practitioners, their opinions could be counted as an affirmative or negative vote on a given issue. The results of this process can be seen in tables 3 and 4.

4.2 Consensus index
For our research, and in order to find the needed consensus among the participants in the interviews, there was a need to obtain a frequency score for each participant on the issue(s) they raised during the interview on the various issues that we wanted to discuss that could be considered as answers to our research. Although there was variety in the roles of the interviewees and the fact that they might not have directly mentioned some clear points on a given issue, the entirety of the responses and their overall reasoning lead us to infer some answers (inductively) to questions that may not have been immediately obvious from the wording of the response.

Our frequency scoring is calculated based on a scale of the number of consensus points mentioned by the interviewee at various points throughout the interview on various issues discussed. The scale is relative because in order to find which issues are more important, practical and/or popular with practitioners their opinions could be counted as an affirmative or negative vote on a topic. Since we are only concerned with the affirmative votes (in the case of consensus), the scale is established to be scored on the basis of the number of affirmative votes on an issue under discussion. The results of this process can be seen in tables 3 and 4.

5. Interpretation of Results
This section of the study attempts to answer the question of “What were the lessons learned?” which try to capture the essence of this idea [13]. What we are looking for here is “a meaning derived from a comparison of the findings with information gleaned from the literature” [14] and our empirical findings. This way we can either affirm the currently accepted information or we could reject them citing new discoveries that we made at this point. To begin with, we will proceed with a systematic discussion of all the major themes discovered as part of our analysis of the interview data.
### Table 3: The Frequency and Consensus Details for RQ1 (Combining Security and Agility)

<table>
<thead>
<tr>
<th>Theme Code</th>
<th>Theme Description</th>
<th>Overall Frequency</th>
<th>Theme Frequency</th>
<th>Interviews Cited</th>
<th>Popularity Index</th>
<th>Positive Responses</th>
<th>Consensus Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ1A</td>
<td>Dedicated Security Engineer</td>
<td>138</td>
<td>30</td>
<td>12</td>
<td>21.7%</td>
<td>25</td>
<td>83.3%</td>
</tr>
<tr>
<td>RQ1B</td>
<td>Software with Security in mind</td>
<td>138</td>
<td>23</td>
<td>11</td>
<td>16.6%</td>
<td>23</td>
<td>100%</td>
</tr>
<tr>
<td>RQ1C</td>
<td>Security Controls</td>
<td>138</td>
<td>18</td>
<td>4</td>
<td>13.0%</td>
<td>17</td>
<td>94.4%</td>
</tr>
<tr>
<td>RQ1D</td>
<td>Informal Security Experts</td>
<td>138</td>
<td>14</td>
<td>8</td>
<td>10.1%</td>
<td>14</td>
<td>100%</td>
</tr>
<tr>
<td>RQ1E</td>
<td>Integration Risk</td>
<td>138</td>
<td>13</td>
<td>7</td>
<td>9.4%</td>
<td>13</td>
<td>100%</td>
</tr>
<tr>
<td>RQ1F</td>
<td>Experience of Developers</td>
<td>138</td>
<td>13</td>
<td>6</td>
<td>9.4%</td>
<td>13</td>
<td>100%</td>
</tr>
<tr>
<td>RQ1G</td>
<td>Security in Planning</td>
<td>138</td>
<td>12</td>
<td>8</td>
<td>8.6%</td>
<td>12</td>
<td>100%</td>
</tr>
<tr>
<td>RQ1H</td>
<td>Static Analysis Tools</td>
<td>138</td>
<td>10</td>
<td>6</td>
<td>7.2%</td>
<td>10</td>
<td>100%</td>
</tr>
<tr>
<td>RQ1I</td>
<td>Static Code Reviews</td>
<td>138</td>
<td>5</td>
<td>2</td>
<td>3.6%</td>
<td>5</td>
<td>100%</td>
</tr>
</tbody>
</table>

### Table 4: The Frequency and Consensus Details for RQ2 (Change Agile Practices for Security)

<table>
<thead>
<tr>
<th>Theme Code</th>
<th>Research Question Theme</th>
<th>Overall Frequency</th>
<th>Theme Frequency</th>
<th>Interviews Cited</th>
<th>Popularity Index</th>
<th>Positive Responses</th>
<th>Consensus Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2A</td>
<td>Agile vs. Traditional Methods</td>
<td>92</td>
<td>27</td>
<td>11</td>
<td>29.3%</td>
<td>20</td>
<td>74.0%</td>
</tr>
<tr>
<td>RQ2B</td>
<td>Awareness of Security to Agile</td>
<td>92</td>
<td>26</td>
<td>11</td>
<td>28.2%</td>
<td>26</td>
<td>100%</td>
</tr>
<tr>
<td>RQ2C</td>
<td>Impact of Accelerated Schedule</td>
<td>92</td>
<td>21</td>
<td>10</td>
<td>22.8%</td>
<td>17</td>
<td>80.9%</td>
</tr>
<tr>
<td>RQ2D</td>
<td>Reduced Security for Internal Projects</td>
<td>92</td>
<td>11</td>
<td>6</td>
<td>11.9%</td>
<td>11</td>
<td>100%</td>
</tr>
<tr>
<td>RQ2E</td>
<td>No Change Necessary</td>
<td>92</td>
<td>7</td>
<td>5</td>
<td>7.6%</td>
<td>7</td>
<td>100%</td>
</tr>
</tbody>
</table>

### 5.1 RQ1- combining security and agility

In order to be able to seamlessly combine security and agility and evaluate the suitability of security mechanisms for use in Agile, the best and most effective practice turned out to be RQ1A the inclusion of the security engineer (18.1%) even though it came at a relatively higher cost than the second most agreed upon theme which was RQ1B developing software with security in mind (16.6%) which did not include the added cost of having to hire a dedicated security engineer for the team. RQ1C involved the use of security controls in mitigating the need for security but the fact that Standard Security Controls are not yet widespread it could make it relatively more costly than other more popular ways of adding security to Agile projects. RQ1D underscored the fact that while there was no formal security engineer, such a skill was still sought after by various team members and more often than not such a person was unofficially picked to help on certain security matters. While this theme is not presenting a solution or a way to combine security and agility, it reinforces the need for RQ1A which is to have a formal dedicated security engineer present. RQ1E came as the only negative set of responses to the question in which the various aspects of the combination was scrutinized. While there are indeed risks and challenges associated with any such undertaking as combining security and agility in a seamless manner, the less than 10% negative response (in terms of risk, not how it could not be done) to this idea shows that there are no major consensus on how and why this method could not be realized. RQ1F shows how the relative experience of the people in the development team has a direct impact on the overall security of the project and how this in and of itself could be used as a way to increase the security of the resulting software for a given project but since relatively few respondents mentioned it as an important factor (9.4%) we can conclude that this pales in comparison to the combined agreement of the benefit of using a security engineer combined with the need for such a person (RQ1A+RQ1D) which is 28.3%.

The addition of the security engineer was simultaneously the most popular and the most agreed upon and contested theme of the interviews. While practitioners recognized and agreed that the addition of the security engineer could benefit the team and the resulting software in significant ways in terms of security they voiced concern in terms of cost as well as the relative impact that the addition of a new member to the team could have on the overall velocity of the project. The security engineer could substantially improve the team’s
overall knowledge of security over time through sharing their knowledge and experience with the rest of the team. They could bolster the security assurance argument of the agile project and mitigate the risks voiced by some researchers and practitioners regarding the Agile principles in meaningful ways. With everything considered, the addition of the security engineer to the team appears to be the best overall practice that an organization can do in order to increase the security assurance argument for their respective clients as well as for their internally or externally deployed software.

The second most popular theme involved the consideration of security at each step of software development without having to formally be required to follow certain steps. This approach helps agile in a sense that the process of agile is not changed or modified as a result of this increased consideration and it will ultimately result in a much more secure software and an increased security assurance argument for the project. This however required increased levels of knowledge and expertise from developers and such people are relatively hard to find in practice compared to a regular, average developer. However, since agile is known to allow tacit knowledge to disseminate throughout the team all developers do not have to be of such high knowledge level and expertise and therefore could train other less experienced developers as they are doing their normal day-to-day activities. This is a lower cost alternative to having a security engineer present permanently as discussed earlier.

Some interviewees were enthusiastic about the use of security controls within software as an alternative to either a security engineer or any other proposed method for adding security to the resulting software. The addition of these controls alleviates the developer from having to come up with and implement many routine security tasks within software that would otherwise need to be implemented manually and the upside is the lower cost and time that could be used on other tasks. The downside of this approach could be that there are no standard security controls at this point but only some proprietary security controls are provided by software security firms. Surprisingly, this solution was not proposed through the literature and only became apparent after a few interviewees expressed in multiple places that standard controls could be a useful, low cost alternative to other more involved and popular approaches.

While the informal security expert is not cited or proposed as a solution to the problem of adding security into Agile, many interviewees mentioned the existence of an informal security expert who could help them on such issues within their reach in the organization. The experts ranged from Software Engineers to Architects to System Engineers and team leaders who were known for their experience and knowledge on such issues. This represented the minimum that they needed in terms of security of the software but in a small way contributed to the overall security assurance for the project. This clearly shows that the existence of a security engineer or expert is indeed required to a certain extent but the actual measure of the degree of such a need and their effectiveness as of this writing is unknown and ambiguous. A take away point from this theme is that the choice of the informal security expert was guided in part by the relative knowledge and experience of the team member who was designated informally to act as a kind of security consultant.

The theme of Integration risk involves all the relative issues and risks that could come up as security is integrated into Agile. While this was a negative theme where the participants discussed drawbacks and other impacts of combining security and agility, the actual frequency turned out to be less than 10% of all the respondents. What’s more, about half of the interviewees raised some kind of issue that could come up as one would combine security and agility for a given project. For example, manager encouragement and support of developers in terms of security is cited as very important in a successful integration. However, since most managers are not concerned much about security for the project their cooperation and setting of the tone for the team in terms of security is a risk that need to be addressed before a successful integration could be accomplished.

RQ1G is another emergent theme that was not mentioned explicitly throughout the literature which involves the developers within the team. While the process of software development involves more than just the developers alone, they are the ones who actually write the software and inevitably introduce the vulnerabilities into the code as well. Knowing which developers are more effective and secure has been given as a focus area by many interviewees both directly and indirectly. The overall consensus seems to suggest that the experience and the level of knowledge of developers directly impacts the resulting software as it relates to security in addition to quality and functionality. Not only they could identify more issues earlier but they could also solve problems and security issues easier than any other member of the team when it comes to increasing the security level of the software under development. While it may not be practical to hire only experienced developers, actually how many are needed within the team to make a lasting impact could be investigated further.

While RQ1G was discussed in the literature as a relatively important issue for security in agile, the data gathered from practitioners indicates less emphasis on its importance but more emphasis on the manner in which planning is done in Agile. Some participants argued that the planning process could include steps towards increased security but the exact manner of changes and how long the process should take was not obvious from the discussions. One take away point from this theme is that the majority of the interviewees view the planning stages of Agile and the lack of upfront design, such as in traditional methodologies, does indeed impact security of the overall software although the exact measure of the impact is as of yet unknown.

While the static analysis tools and similar software programs indeed help in finding threats and vulnerabilities
in the code, the actual practical value of using them in the project is relatively unknown since less than 30% of the participants even mentioned static analysis tools. From amongst those who did mention the tools the consensus was positive overall but at the same time they echoed the position that was raised in the literature which was the fact that the static analysis tools are not mature enough to be able to discover many forms of threats and vulnerabilities. Overall static analysis tools do provide added security assurance but not enough to achieve good enough status amongst academic and practitioners alike especially when it comes to using them as a major factor in combining security and agility.

RQ1I was only mentioned because some people felt that the practice of code reviews could be used as a way to augment and increase the relative security assurance of the software but by itself this practice is in no way intended to be used as a standalone solution such as the inclusion of the security engineer to the team. Furthermore, in order for this practice to be effective the reviewer must be more experienced and knowledgeable than the other developer who wrote the code. For example, code review during code check-in time, a review by an informal security expert or a more experienced developer and/or software engineer could help in pointing out certain defects and vulnerabilities which will also result in increased awareness and knowledge of security for the less experienced developer as well.

5.2 RQ2- changing agile practices for security
To be able to better understand the overall issue of whether or not changing agile for the sake of security really needed and to see what kinds of projects could use security more than others, the responses gave us a relative scale to interpret the findings. Out of all responses to the questions that were related to this research question, 59.7% of the responses agreed on how the changes to Agile were not really necessary from 4 overall points of view (themes). From those who were in favor of changes that agreed with no changes to Agile being necessary the themes involved were RQ2A, RQ2C, RQ2D, and RQ2E. We also gained insight into what projects could use security more than others and it came to whether or not the project was being used internally within the organization or in a more public setting open to scrutiny by everyone.

RQ2A was one of the most discussed topics when it came to changing Agile for the sake of security. Some participants stressed that the existence of security was independent of either methodology (Traditional or Agile) that should be irrelevant whether security should be included or not. However, the majority of the participants agreed that Agile was not less secure if not better in terms of security and some went as far as claiming that Agile was in fact better in terms of quality and adapting to changes which allowed it to find and resolve more vulnerabilities faster and earlier than traditional methodologies. This coupled with the fact that traditional methodologies by design are not open to change provides the insight which is to say that Agile overall does a better job of developing more secure software than waterfall even without the inclusion of extra security focused steps and therefore does not need to be extended for the sake of security.

Awareness of security in Agile turned out to be almost as important as the discussion of Agile vs. traditional methodologies. However, this was one theme that almost every participant talked about which underscores the need for everyone involved in the project to be at-least aware of security when they design and develop software. If security is important in any way to the project, then stakeholders must think about risk assessment and analysis, elaboration phases, upfront planning and investment, policies and guidelines geared towards increasing every team member’s awareness of security and allow them to voice their concerns freely and openly. This suggests that more focus on security is needed that could indeed be accomplished through the addition of certain steps and/or practices to Agile.

RQ2C was an emergent theme which discussed the accelerated timeframes of Agile and their effects in practice especially for security. While some participants characterized scheduling issues as irrelevant to security the majority of the respondents agreed that the accelerated timeframe often employed in Agile projects tended to be too short and this had a cascading effect on all aspects of development and especially affected security because security is considered as a secondary objective to functionality which is a primary concern of stakeholders in Agile. Since security is classified as a cross cutting concern and often results in non-functional requirements there is less perceived value in its implementation and therefore the added schedule pressure often forces people to cut corners in this area. This is not to say Agile as a methodology is driving this issue but it is the people who choose too short of a timeframe per iteration that results in too much pressure which results in more risk than is warranted. This theme underscores the need for people in charge of creating plans and schedules to be more open and conscious of security related issues and dedicate more time per iteration to address other issues in addition to functionality.

RQ2D was an emergent theme (not mentioned in the literature) that many participants discussed and some even had direct experience on which provides an insight into how internal projects are handled and undertaken in terms of security. Many participants claimed that they did not focus so much on security because of the fact that their software was only deployed internally within a controlled and trusted environment which was either already being protected through the infrastructure or through other means such as various control mechanisms deployed throughout the organization (Access Control). They also mentioned a dedicated team sometimes being present that was handling security issues within the organization and that the software was less prone to threats and vulnerabilities as a result. This further shows that for some projects the security aspect simply is not a concern
because either it is being handled at a different layer or the trusted and private nature of the deployment environment minimizes any chances of exploitation and misuse.

RQ2E was a theme directly taken from the literature which was also talked about by some participants. They basically agreed with the premise that no change was indeed necessary to Agile and that the addition of security should be handled as much as possible through unobtrusive means that do not attempt to modify or extend Agile with additional steps and/or practices that may not be needed or required by all projects. This is very important because it clearly shows that there are certain numbers of people in practice who feel they need to keep Agile the way it is in order to keep benefitting from its advantages.

6. Sampling and Design Issues

According to Boyatzis “Thematic analysis is sensitive to the quality of the raw data or information. Therefore, sampling decisions not only affect but to a large extent determine the degree of reliability and validity attainable” [11]. The quality of the criterion selection and the sampling will determine the quality of the code and subsequently the quality of the findings. Our criterion for targeted sampling consists of multiple criteria which includes roles of various groups of stakeholders within a software development team such as Developers, QA, Managers, Security Experts, Consultants. The sampling frame consisted of predominantly developers including programmers, architects, and software engineers, some Managers, Security Experts, QA/Testers, and Consultants. Sub-samples include 10 Developers, 6 Managers, 3 Security Experts, 4 Consultants, and finally 3 QA/Testers. These individuals were at the time working in various organizations that develop software and since they could not be reached in their natural work environment, they were pursued in Conferences, periodically held professional meetings, at various lectures and events, and at other similar events. An important aspect to note is that there is some overlap between the roles of subjects for example some developers were also managers and some other developers were also consultants and we elected to count both roles for that individual.

After conducting these interviews we became confident that we had interviewed enough individuals because the nature and types of answers that they were providing us was the same as other previously interviewed subjects. At the same time, we had achieved the goal of being comprehensive in sampling by obtaining information from various roles within the agile team that we were targeting for the interviews which according to Boyatzis (1998) gives the most complete picture of the unit of analysis which in our case is the security issues within the Agile team. After interviewing the last 5 individuals we were assured that there was not much more new information we could gain from continuing the interviews with more people and therefore we stopped interviewing individuals after the 15 interviews were concluded.

7. Reliability

The aim of establishing reliability is to be able to show consistency of approach in how the process was conducted [15]. According to Yin, the procedures of the study need to be as detailed as possible in order to increase the understanding of the other researchers and readers about how the qualitative study was conducted [16].

7.1 Reliability threats and biases

We took steps to make sure all participants are properly chosen based on their background and their relative work experience in the area of our research and a variety of roles and responsibilities were chosen for the semi-structured interview in order to minimize any bias by one particular group over an issue. We chose the format of Semi-structured interviews so that the questions could be posed from different perspectives and different people while the answers would contribute to the same basic research question.

We took extra measures in keeping the anonymity of the respondents for their interviews. We went as far as anonymizing their organizations and erasing any personally identifiable information that might possibly be used to trace back to them in any way. The developers might be biased towards their role as instrumental in the process of adding security to the software therefore they might have an elevated sense of ownership and might want to exaggerate their achievements in order to appeal to a higher sense of technological ability. Our analysis took this and other such biases into account and only considered the responses credible if they were corroborated by other participants with different roles from different organizations in various countries and cultures and genders.

We recorded interviews in order to minimize observer error (in transcription) and later went back to the recordings and summarized the findings as accurately as possible in order to minimize any possible observer errors. On the same note the structured format of the semi-structured interview provided us with an added assurance that the same exact wording for each question would be asked from all respondents therefore reducing the possibility of this threat to reliability.

To ensure we have interpreted the replies in an unbiased way, we have used the analysis software package NVivo in order to provide a more objective method of interpretation of our results in order to minimize observer bias in our results. We also send the results back to the participants if/when they asked us to do so in order to confirm their responses are still valid and accurate and they were given another opportunity to change/modify any previous answers.
8. Validating the Findings

Validity means whether or not the findings really represent the results that were present. According to Saunders, Lewis et al. (2007) validity is “Concerned with whether the findings are really about what they appear to be about” [17].

8.1 History

For this threat to validity, the researchers must ensure that there have not been any recent events that might undermine people’s opinions about the subject of your research. We have interviewed various stakeholders from different roles and background as well as different geographical locations in order to minimize any local, historical bias from entering into our results through artificial means. Additionally we did not encounter any case where the responses of the participant(s) were not truthful or a case where they inaccurately represented their position within their respective organizations.

8.2 Testing

Testing says if the respondents feel their answers might in some way affect them after they have been interviewed this might affect the results. To guard against this threat, we chose to conduct our interviews in a completely anonymous fashion so the respondents may feel free from any pressure (job related or otherwise) to be able to answer our questions freely and honestly. Even with the assurance of anonymity we still encountered some reluctance by various people in sharing specific job related information. In some cases only after the recorder was turned off they started to speak about certain issues they were having that they were afraid to share formally.

8.3 Instrumentation

If the respondents’ circumstances were such that it affected their results, we ensured that none of our questions were time and/or place dependant in order to minimize the chance of any unforeseen circumstances to affect the respondents’ answers to our semi-structured interview questions. For example, people who opted for phone interviews tended to spend more time answering the questions perhaps because they were not under any time/scheduling pressure and were feeling more comfortable. On the other hand, meeting face-to-face allowed the interviewer to establish a greater degree of trust and therefore the participants were more open to answer questions in detail.

8.4 Maturation

This threat is concerned with the question of: does any environmental and other events could possibly affect the participants’ responses indirectly? In our research, unless something were to happen during the interview itself that might prompt a participant to change their answers, for example their boss walking into the interview meeting, there is no conceivable way that we could think of that could affect our results. On the case of the boss being around, we ensured this would not happen by scheduling the participant at a date and time that is most suitable for them either on the phone or face-to-face in a public setting. We did encounter one case where the potential participant and their boss where in the same area and that made the potential participant hesitant and reluctant to answer questions freely.

8.5 External validity

External validity is concerned with whether or not the results apply to settings other than those that was conducted by the researcher, such as other organizations, etc. Since we have chosen our participants from various roles and diverse geographical locations, we have tried to ensure that our results are applicable to as many software development firms as possible. These results would ideally be applicable to any team within any size organization that is of small to medium size which is following one or more Agile principles and/or methodologies. Furthermore the more experienced the team member the better insight they would have into the issues and discussions involved so the relative accuracy of each interview cannot be compared but the collective responses given by all participants should be consistent with a similarly chosen sample from the same diverse population of practitioners.

8.6 Logic leaps and false assumptions

These set of threats discuss Data Collection issues: “Is it logical to assume the way you are collecting your data is going to yield valid data?” [17]. Since we are collecting our data in an entirely anonymous way, we have allowed for the highest degree of transparency from the respondents without the possibility of any backlash or retribution therefore we feel our data collection is as valid as one could reasonably expect and the number of interviews and consensus that we have reached further suggests that we have indeed been getting consistent and valid results. To ensure that we progressed from a large amount of data to our conclusions in a coherent and structured (using a theoretical framework) fashion, we followed a hybrid approach of inductive and prior research driven framework.

8.7 Negative responses

Even though the process of semi-structured interviews was completed successfully and we gathered the required amount of information for each interview, there were instances where we were faced with inconsistencies and additional concerns despite our assurances of anonymity and confidentiality. For example, when we asked a participant “what are your project or code level security practices that are in use today in your organization?” we got the answer of: “I have to tell you I cannot answer some specific questions for the company I work for”. This is because the question was about security. Only after we asked the same question from different perspectives but more subtly was when we got relative answers but in general some people exhibited an automatic negative
response to sharing any kind of security related information anonymity or not which underscores the sensitive nature of the topic under investigation.

9. Conclusion and Further Research
This paper concludes the investigation into security issues in Agile which allowed us an in-depth look into the various details from multiple perspectives and points of view. Each major theme in our findings can now be investigated further in order to gain a more specific and targeted information about how each proposed solution may be useful in a real world setting and we urge more research to be conducted in these areas to shed more light on the merit of each proposed solution.

References