Tales of Software Updates: The process of updating software

Citation for published version:
Vaniea, K & Rashidi, Y 2016, Tales of Software Updates: The process of updating software. in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.* ACM, pp. 3215-3226, Computer Human Interaction (CHI) 2016, San Jose, United States, 9/05/16. https://doi.org/10.1145/2858036.2858303

Digital Object Identifier (DOI):
10.1145/2858036.2858303

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems

Tales of Software Updates: The process of updating software

Kami Vaniea
The University of Edinburgh
United Kingdom
kvaniea@inf.ac.ed.uk

Yasmeen Rashidi
Indiana University
Bloomington, IN, USA
yrashidi@indiana.edu

ABSTRACT

Updates alter the way software functions by fixing bugs, changing features, and modifying the user interface. Sometimes changes are welcome, even anticipated, and sometimes they are unwanted leading to users avoiding potentially unwanted updates. If users delay or do not install updates it can have serious security implications for their computer. Updates are one of the primary mechanisms for correcting discovered vulnerabilities, when a user does not update they remain vulnerable to an increasing number of attacks. In this work we detail the process users go through when updating their software, including both the positive and negative issues they experience. We asked 307 survey respondents to provide two contrasting software update stories. Using content analysis we analysed the stories and found that users go through six stages while updating: awareness, deciding to update, preparation, installation, troubleshooting, and post state. We further detail the issues respondents experienced during each stage and the impact on their willingness to update.

Author Keywords

Software Updates; Human Factors; Security

ACM Classification Keywords

D.2.7. Software Engineering: Distribution, Maintenance, and Enhancement; H.5.2. Information Interfaces and Presentation: User interfaces–user-centered design; K.6.5. Management of Computing and Information Systems: Security and Protection

INTRODUCTION

Software updates provide modifications to existing software, but some updates are more likely to be installed than others [?, ?]. Updates change software, fixing known bugs, changing features, and altering the user interface. Some changes are welcome, even anticipated, and some changes are unwanted making updates simultaneously exciting and risky for end users. In this work we survey respondents about two contrasting software update experiences in order to understand what aspects of the update process encourage or discourage updating.

Looking at publicized update failures it is easy to understand why some people might associate updates with bad effects. Recently Microsoft was accused of adding Windows 10's "spy" telemetry features to Windows 7 and 8 through an update [?]. On personal devices updates can be seen as annoying and time consuming to install. Senator John McCain once asked Apple CEO Tim Cook "why the hell [do] I have to keep updating the apps on my iPhone all the time and why you don't fix that?" [?]. The changes updates bring can also be unwelcome to users who liked the way their systems used to function or need to use specialized accessibility software.

Installing updates is an important part of security maintenance which is not obvious to users [?]. One of the best ways to protect a computer from malicious software is to install security related updates in a timely manner [?, ?, ?, ?, ?]. The majority of computer compromises result from vulnerabilities where an update is available that corrects the vulnerability but has not yet been installed [?, ?]. Malicious software targets machines with open vulnerabilities, using them to gain access to important parts of the operating system. Updating quickly is also important. As soon as a vulnerability becomes public knowledge, exploit rates jump by as much as 5 orders of magnitude [?, ?]. Installing security updates closes vulnerabilities preventing attacks from being successful. Systems that are regularly updated have both smaller attack surfaces and less compromise attempts [?, ?].

Prior work has shown that people do not always understand why updates are necessary or what they do [?, ?] which can lead to a decision to avoid updating software when the update is perceived as not needed [?]. This situation results in a dead weight loss where users might have preferred the new version, and developers would prefer to maintain fewer versions, but the user is not updating due to potential risks and unclear benefits.

In this work we are interested in the aspects of the software update process that are the most salient to end users. We want to know what people remember from prior update events. Particularly, aspects that caused them to want to install an update or caused them to avoid installing an update.

We conducted a survey with 307 respondents recruited from Amazon's Mechanical Turk where we asked them to provide two contrasting free text update event stories. Using con-

CHI'16, May 07 - 12, 2016, San Jose, CA, USA

ACM 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858303

tent analysis, we analysed the stories and found that users go through six stages while updating: awareness, deciding to update, preparation, installation, troubleshooting, and post state. We detail the types of issues respondents encountered during each stage that impacted their willingness to update.

BACKGROUND

The impact of updating software on security is not obvious to end users. Fagan et al. showed that people have difficulty understanding the relationship between software updates and computer security. People are also hesitant to apply updates because they are annoyed or confused about the update message that they received [?]. Vaniea et al. similarly found that users avoided application updates due to: 1) unanticipated user interface changes, 2) unused and unrecognized software, and 3) liking the current software [?]. Ion et al. compared the security advice of experts and non-exerts, they found that 35% of experts mentioned installing updates as one of the top three things they do to stay safe, while just 2% of non-experts made the same recommendation [?]. Tian et al. studied mobile application updates (apps) via a survey, they found that nearly 60% of users had previously decided to not update an app [?]. They also tested a new notification interface that highlighted update-related comments.

Our work is the first to explore the full process of updating from the prospective of an end user. While some earlier work has looked at issues users experience most of the earlier studies are either small [?, ?] or focus on very narow aspects of the update process [?, ?].

Automatic updating

While keeping the user informed is important, it is not always necessary to keep them in the loop for all security decisions. If the correct action is known, then it may be safe to make a decision without user involvement [?]. One obvious approach to improve update compliance is to automate update installation. Microsoft has already shown this approach to work. In Windows XP SP2 they enabled automatic update installation and saw installation rates jump from 5% of SP1 computers to 90% of SP2 computers [?, ?]. Wash et al. found that with the exception of the auto-reboot feature, Microsoft's automatic updating mechanism was helpful to users and kept their machines safer than they might otherwise have been [?]. Internet Explorer, Chrome, and Firefox all now support silent automatic updating.

Automatically installing updates without user intervention improves installation rates and security [?], but the practice has three major limitations. First, updating software can cause compatibility issues with older or proprietary software [?], so users need the ability to turn off automatic updates. This is a major issue in companies where in-house software depends on specific versions of software such as Java or Adobe Reader. Users with disabilities also need to be able to disable updates until their accessibility programs gain comparability [?]. Second, users have the right to decide what software gets installed on their machines [?]. When an automatic update silently does something unexpected and unwanted it may result in people feeling betrayed, causing them to loose trust in the automatic update process [?, ?]. Vaniea et al. observed 33% (8 out of 24) iTunes users stop updating after an unanticipated user interface change, even though the subsequent updates made no user interface changes [?]. Third, when users are not involved in the update process it becomes harder for them to build good mental models [?, ?]. Without these models users have trouble understanding what is happening as well as how to control it [?].

Warnings

Update requests have some similarity to warnings in that they are notifying the user about out-of-date software and asking the user to take action. Unfortunately, users do not always understand or heed the warnings presented to them [?]. SSL warnings in particular have this issue [?, ?, ?, ?] as do Firewall warnings [?]. Users become habituated to ignoring warnings [?, ?]. Work by Bravo-Lillo et al. has looked at ways to counteract habituation through user interface designs that encourage the user to interact with the important information on the warnings [?, ?, ?, ?]. His work shows that interaction with the important information on a warning can improve people's comprehension of the warning content.

METHOD

Data was collected using a survey instrument and then analysed using content coding. The survey design puts emphasis on obtaining both positive and negative update stories to get a balanced understanding of why people do update as well as why they do not update.

Survey instrument

Inspiration for the survey design came from Rader et al. where they asked survey respondents for stories about computer security experiences [?]. Some of the questions in this survey are directly drawn from their work.

The survey started with 12 demographics questions including age, nationality, education, technical experience, and type of computer commonly used. We followed that with an open-ended question that asked the respondents to share with us an update-related experience:

> Please share with us an update related experience. This can be any experience you have had while updating software on any device such as a phone, game console, computer, or tablet. Or an experience where you decided not to install an update. This can be any event involving an update such as the last time a piece of software asked you to update it, or when you noticed that your software had changed due to an update.
>
> Please select an update experience for which you can most easily recall details about where you were and what happened when you installed (or chose not to install) the update. You will be answering further questions about this experience in the next two pages. In couple of sentences please summarize what happened in your own words.

We then asked 17 structured follow-up questions about the details of the story. The questions had a three way branch based on whether they had installed the update, did not update, or could not remember. All branches asked similar questions but with different phrasing and options. The follow-up questions asked respondents to characterize their story as a negative, positive, or neither negative or positive experience.

After the follow-up questions, we asked participants to share a second contrasting story with an opposite characterization. If they characterized their first story as positive, they were asked to provide a negative second story. If the first story was neither positive or negative they were just asked to provide another story. Respondents were informed that no follow-up questions would be asked about the second story.

Finally, we asked seven likert questions about general update opinions. In this work we focus on the two stories and do not discuss the contents of the last seven questions.

Respondents

Respondents were recruited from Amazon’s Mechanical Turk. The survey was advertised as “15 minutes survey on software update experiences” and pilot tested to ensure that 15 minutes was an accurate upper-limit estimate. We required that MTurk respondents be located in the United States and have a task approval rate of at least 95%. Participants were compensated $1 each for completing the survey.

We had 307 completed responses. Of those, 43.3% were female with ages ranging from 18 to 74, average of 35 years old with a standard deviation of 11.4. Participants had different levels of education with 55.4% (n=170) holding a Bachelor’s degree or higher. We asked respondents what devices frequently use, the majority of respondents 71.7% (n=220) use Windows computers, while 12.4% (n=38) use Mac computers, 9.8% (n=30) use smart phones, 2.9% (n=9) use tablets, 2.6% (n=8) use Linux computers, and 0.7% (n=2) use other devices. While the majority of respondents reported that they have never worked in any “high tech” job as computer programming, IT, or computer networking, 73.3% (n=225), over a quarter had worked in a high tech job. Generally we found that people with more technical experience had similar issues to those with less experience, though the language they used when describing the issues tended to be more detailed.

Code book design

Based on the outcomes of prior research [**?**, **?**, **?**] we initially assumed that users’ update concerns would center on two issues: temporary loss of state (i.e. reboots, lost work, and setting changes) and post-update changes to the software (i.e. user interface, performance, and features). The initial code book was constructed using a deductive methodology based on the earlier results. Using this initial coding scheme one member of the research team tried to code a random sample of forty story pairs (eighty stories, two per respondent). In their opinion, the conceptual framework of the deductive codebook was missing several of the key concepts respondents were attempting to express.

The deductive code book was therefore abandoned and we instead focused on constructing codes using a more grounded approach that encouraged us to use the language and concepts expressed by respondents. One researcher used the In Vivo coding methodology [**?**, **?**] to code a random sample of forty story pairs. The In Vivo coding methodology was selected because the coder uses the respondent’s words to code elements from each story. For example, P30 said: *“I think it’s important to wait a while for Apple and beta testers to find and work out all the bugs often associated with these updates.”* [*P30S1*] which might have the codes “beta testers” and “work out bugs” associated with it. It is important to note that while this coding style can be highly subjective, its strength is that it encourages the coder to phrase concepts in the respondents’ language.

The In Vivo coding identified many issues similar to the expected concerns described in related work such as resource usage and reboots. The key concept missing from the original code book was a way to describe the concepts within the workflow of the installation. When describing an update experience, respondents would focus on different stages of the installation. This observation lead to the construction of four top level codes to describe the stages of an update installation: *Previous* software state, *Initiation* of the update, *Installer*, and *Post* software state. Respondents also described their expectations of the installer and post state leading to two more top level codes of *Expected-Installer* and *Expected-Post*. The *Impact* of the installation on respondents as well as their *Behavior* were also coded. These codes were later refined into conceptual categories which we discuss in findings. The new code book was designed with a hierarchical coding structure. Codes were assigned at the granularity of a single story. That is, if a story described two reboot events during update installation the story would only be coded with one “Installation:Reboot” code.

Two researchers then used the new code book to separately code the same set of ten randomly selected story pairs. Newly identified codes were discussed and the code book was further refined. This step was repeated three times till the coders felt that the code book was comprehensive and the ambiguities between codes were clarified in the code book definitions. The two researchers then selected a random sample of thirty story pairs, coded them and computed a Cohen’s Kappa of 0.90 (Z=67.61, p-value<0.0001) which shows high agreement between coders. The two coders then divided the remaining story pairs and coded them separately. After finishing, a random set of 56 story pairs were selected and coded by a second coder, these were used to compute a Cohen’s Kappa of 0.84 (Z=90.40, p-value<0.0001) which shows a high degree of agreement during coding.

Both coders kept notes during the coding process to record concepts that were not well covered by the code book. One issue that came up involved reasons people decided to not update or delay an update. Many of the stories spanned multiple update events. For example, a respondent might describe how they decided not to update due to concerns, then a new update came out that addressed their concern, so they updated. This mix of update events made it challenging to correlate codes associated with not updating and the reasons for not updating. Using the existing code book and notes taken during coding the two researchers constructed a new set of codes describing reasons respondents avoided or delayed updates. Both researchers then went through and coded all 112 stories focusing only on the part of the story associated with the decision to delay or avoid updating. Inter-rater reliability of this coding had a Cohen's Kappa of .80 (Z=45.80, p-value<0.0001) which shows high coding agreement.

Stories

Participants provided 614 stories. Of those, 14 were judged to be about something other than an update event, and 8 contained no story at all, resulting in 592 valid stories. All respondents provided at least one valid story. In the remainder of the paper we will be referring to the first story in the survey as story1 (S1) and the second story as story2 (S2). When quoting stories we indicate the participant number and story number as: *"Quote."* [P10S1] For readability, quotes have been minimally edited for capitalization and obvious spelling errors such as "teh."

For their first story, 147 (49.3%) of respondents related what they charactorized as a negative experience, 88 (29.5%) related an experience that was neither positive or negative, and 63 (21.1%) related a positive experience. Respondents were concise when conveying stories. Story1 had an average of 58 words with a standard deviation of 34.7 words. The longest story1 had 301 words and the shortest only 11. Story2 had an average of 38 words with a standard deviation of 22.5 words, a minimum of 5 words and a maximum of 195 words.

The first story had minimal priming before it and a set of follow-up questions after it. The second story followed the story1 questions and asked for a contrasting story. Therefore story2 had more priming and was more constrained than the initial story provided. This ordering was an intentional part of the methodology. We were rightly concerned that respondents were more likely to provide a negative experience and we wanted to know what both good and bad stories contained.

FINDINGS

One key observation from the initial In-Vivo analysis of 80 stories was that the update process involved more stages than the obvious notification, installation, and post state. During post analysis we further subdivided and re-categorized the codes to thematically bring together topics that occurred in groups. We find that when installing an update users go through the stages of: *awareness, deciding, preparation, installation, troubleshooting*, and *post state*. The stages and the issues identfied in each are summarized in Figure 1.

User concerns were surprisingly consistent across the update stages, though the context and specifics of the concerns changed. We find that respondents were primarily concerned with four issues: *reputation, resources, bugs, and disruption*. We discuss these issues in terms of the stages of update.

Awareness

We anticipated that notifications and automatic updates would be important aspects of updates, so in the follow-up questions to story1, we asked respondents: "How did you become aware of the update?" The answers are shown in Table 1. The two most common were that the update displayed a notification, and that the update started automatically installing without user interaction. Given the shift towards automatic checking for and installing of updates the prevalence of these answers makes sense. However, an impressive 63 respondents indicated that they became aware through non-automatic means, including manually checking, being told about it by a person, and being told by another piece of software. The story contents mirrored these results.

Became aware of the update when:	Count
Pop-up asked me to update	164
The update started to install, and I had no choice	41
I thought there might be an update available so I went and looked for it online	27
News, forum, blog, or a friend mentioned the update	20
Another software asked me to do this update	16
I got a notification after the update was installed	6
I noticed that the software changed, so I assumed an update had been automatically installed	3
My computer rebooted	2
Other	21

Table 1. Answers to "How did you become aware of the update?" in the follow-up questions for story1. Respondents were shown the above list and asked to select one answer.

Notifications

Notification dialogs were commonly mentioned as how the respondent became aware of the update. For example: *"My Java notice came up and reminded me to update."* [P207S1] Most stories had a neutral tone towards the notifications but for some respondents the frequency of the notifications were annoying and off-putting.

> *Adobe is very annoying with their update requests. I get a lot of notifications asking me if I want to update and I don't think there's any reason I really need to update the software. It is overbearing and annoying.* [P131S2]

Respondents were also concerned with their ability to distinguish between a valid update notifications and a malicious ones. P95 explains how she had difficulty knowing if an request was trustworthy: *"I honestly could not tell if it was trustworthy. I did not know if I should accept or not. I chose not and it was a wise choice as a friend told me later it was a virus."* [P95S1]

The most common methods of handling the uncertainty were to not update, or to only download updates from the official website of the company. P37 describes how they only download updates from the official site: *"I did it the right way. I downloaded form Mozilla (not a third party site)."* [P37S1]

Automatic updates

Automatic updates were also commonly talked about, though opinions were mixed. Some respondents wished that all updates would happen invisibly without their involvement.

> *I am dissatisfied with any program that does not update itself. Some parts of ASC required me to go to their websites, download new installers, and install the update myself, and that's just lazy coding.* [P196S2]

Others wanted to be in control of their computer and not allow updates to automatically make changes. Stories in this category described situations where an automatic update caused

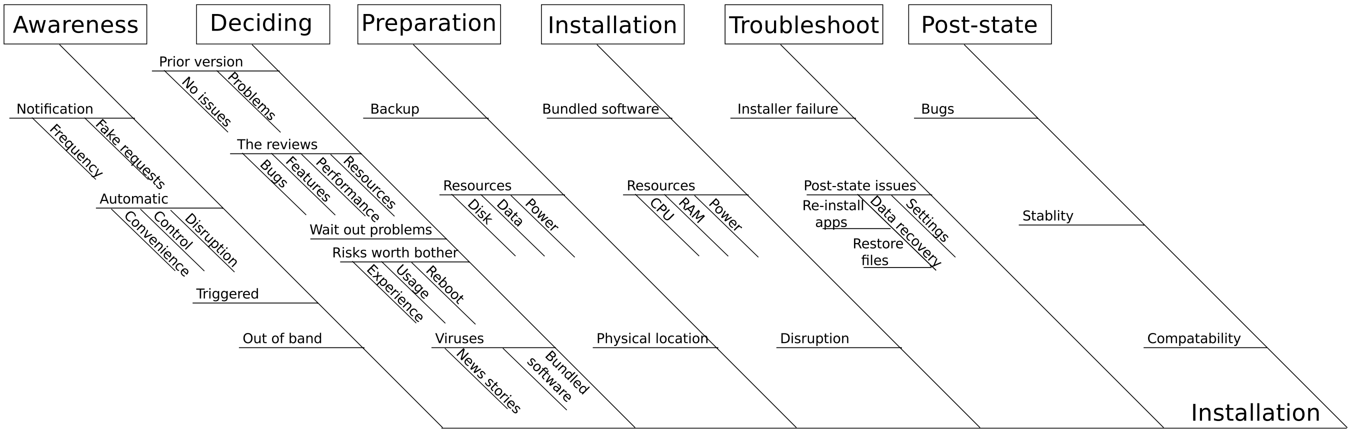

Figure 1. Overview of the stages of updating and the issues respondents experience at each stage.

an unacceptable amount of disruption resulting in the respondent disabling future automatic updates. P241 relates a detailed situation of this type:

> *I have flash installed so I could watch videos. One time my flash stopped working. I found out Adobe had disabled flash somehow in my browser and wouldn't let me use flash again until I installed their new update. I was extremely pissed off that they had disabled something I had installed without my permission so I figured out how to disable Adobe updates in Firefox code and reset so I could run flash again without installing an update. Adobe really makes me mad especially when they try to get you to install spyware with every new update.* [P241S1]

Requirements that devices be rebooted to complete an update caused some respondents to turn off automatic updates.

> *I used to have Windows 8 update automatically, which was quite annoying. It kept on putting it off when it asked to restart, then it asked me to restart and update 6 hours later or something. I was in the middle of a good paying MTurk HIT and the computer restarted automatically. The worst thing is it took about an hour to update, since it was a huge Windows 8 update. By the time I got back to the HIT, it had expired. After that day, I looked up how to turn off auto updating on my laptop.* [P143S1]

Respondents felt responsible for managing automatic update settings and some blamed themselves for not changing the setting when an unwanted automatic update occurred.

> *[Windows Update] closed my system down and rebooted. I ended up losing some work. Its my fault for not turning off updates. I typically like to just set updates to tell me there is updated and lets me read up on them and then applying them.* [P12S1]

Triggered updates

Not all automatic updates install right away, some wait for a particular triggering event to occur before initiating the installation. Microsoft Windows Update is a classic example of this behavior, updates are automatically downloaded and the user is notified, updates are then installed automatically when the user turns off the computer or after a set time period [**?**]. Other updates wait for resources to become available such as a power source or a wireless access point. Respondents who were cognizant of these triggers were able to use them to control when their update would install.

> *My cell phone ... had a system update. This update would automatically occur when I turned on the Wi-Fi on my phone. I did not want to have the new update on my phone, as I heard there were some negative effects. I had to research how to force stop the app, and I decided to not allow the install to occur.* [P107S1]

Out of band

Finally, some users became aware of the update through other means. One of the more common was to look for an update manually as part of troubleshooting an issue. P232 was trying to debug a battery issue: *"After 2 weeks of use [my Surface] started holding less charge. I updated the bios, which seemed to help slightly."* [P232S1] Other respondents heard about an update through channels beyond their computer. This was most common with upgrades where respondents learned about an upcoming iPhone upgrade, or a new version of Windows through the news rather than through a computer notification.

Deciding

Once aware of the update, the user had to decide if they were going to install right away, delay, or avoid the update entirely. In the follow-up questions to story1 we asked respondents if they had or had not installed the update in their story and why (Tables 2 and 3). The options were derived from a combination of our own [**?**, **?**] and other's prior work [**?**, **?**].

The two most common reasons to install an update were that the respondent always installs updates and they thought the update was important. Only 61 respondents indicated that they had not installed the update. For them the most common reason was that they were satisfied with the current software.

Only 25 stories described a situation where the respondent stopped updating all together. More frequent (n=88) was the decision to either delay or to skip a particular update in favour of future updates. P86 describes an update that caused other

Reason	Count
I always install updates	118
I thought it was important	109
I trust this software company	90
I use the software frequently, so keeping it updated is important	87
I didn't have a choice	58
It was a security related update	49
I searched for the update and installed it myself	15
I wasn't satisfied with the current version	15
The current version was broken	14
My family and friends recommended it	6
Other	11

Table 2. The 242 respondents who in their first story indicated that they did install the update were asked to select one or more of the reasons shown above.

Reason	Count
Satisfied with the current version	17
It looked like it would be disruptive	13
I didn't trust the update	13
Compatibility issues	13
Had trouble updating	11
I didn't think it was important	10
I didn't want to lose stuff while updating	6
Too many updates for this software	6
I don't install updates	4
I don't use this software	4
My family, friends, or colleagues didn't recommend it	4
I didn't have time	2
Other	9

Table 3. The 61 respondents in their first story who indicated that they did not install the update were asked to select one or more of the reasons shown above.

people problems, but he wanted to install: *"I have stayed away from the update for now. I continue to look at reviews to see if the update has been fixed or not."* [P86S1] Installing an update was viewed as a way to improve the software. Even after a negative update experience respondents were open to installing future updates, though they might be more inclined to research the update first.

The old version had problems

Respondents who updated tended to focus their explanation on the prior state of the software such as it crashing, not working, or negatively impacting other software. P272 describes why he updated: *"I was having issues connecting to the internet on my laptop and was disconnecting from the WiFi network every 1-3 minutes."* [P272S1] Most discussions around problems were also associated with positive or neutral stories. Only 3 out of the 31 stories mentioning pre-existing problems were marked as negative stories by users.

The reviews said the update would be bad or good

Respondents valued the opinions of other people who had installed the update before them. They asked their friends, read forum posts, read comments, or generally tried to discover if the update was worth installing. If other people did not recommend it, or if their research suggested a feature they valued would be negatively impacted, they did not install the update.

In the follow-up questions for story1, respondents were asked: "Did you attempt to learn about the update by reading information online before installing it?" 32% of people who installed the update researched the update in advance and 46.6% of people who did not install the update researched it in advance. Respondents who avoided updating due to reviews tended to view the update experience as positive. They were happy that they had avoided a negative experience.

Nearly all the stories that involved researching and not updating had a similar outline, the respondent had learned that the update would have some undesirable behavior and had therefore decided not to install it. P307's story was typical: *"Just this morning my Instagram app offered an update. I had heard bad things about the update and I decided not to install it because of this."* [P307S1]

When discussing the "bad things" that the update was reported to do, respondents tended to focus on four points: features, performance, resources, and bugs.

Features were any capability of the software the respondent particularly cared about. Often respondents simply said that the update had "worse features." P244 describes not installing due to a potential feature loss: *"I had an app that would allow you to stream and record YouTube videos. The last update that was sent out removed this capability. Due to this removal I did not install."* [P244S2]

Performance and "speed" were also serious concerns for users, particularly those using mobile devices. *"I decided that I wasn't going to install the update because I have heard all the reviews online about how it generally makes your phone slower in every aspect."* [P68S1]

Resources concerns were similar to performance, but here respondents tended to focus on disk space and battery usage. *"Everyone was saying that the update would drain my battery life quicker. So I avoided doing the update until the next update afterwards."* [P193S1]

Bugs or problems with the software were generally seen as unintentional issues that developers were likely to correct in future update releases. If reviews said that the program had bugs the respondent was likely to skip or delay the update. *"I read on [the software's] support forums that an update was causing certain features of the program to break, and they were waiting to release a patch. I elected to not update the software."* [P89S1]

Wait out the problems

Not all users were willing to spend the time needed to research what an update would do, but at the same time they were reluctant to accept the risk that the update might be bad. To balance the risk and rewards they instead delayed all updates under the theory that by the time they updated other people would have found and fixed all the problems. P30 talked about how he waited for the "beta testers" to find all the problems: *"I think it's important to wait a while for Apple* *and beta testers to find and work out all the bugs often associated with these updates."* [P30S2] P94 described using a similar heuristic with Windows: *"I got a notice to update Windows, and I delayed it like I always do. I delay them for a few days to make sure no bugs are reported with the updates."* [P94S1]

Are the benefits worth the bother?

Disruption was a major point of concern when deciding to install or delay an update. Respondents wondered if benefits of updating were worth the effort required to update and the risk of unwanted changes.

Based on prior update experiences, respondents formed opinions about what the update was likely going to do. These opinions were then used when deciding if they wanted to install the new update. P225 talked about what he expected the update experience to be like: *"Every time I update my Riot LoL, it takes way too long, I feel like my computer gets slower, and there is a new bug in the game."* [P225S2] In short, he thought that the update would be disruptive and not improve the software, leading to a disinclination to update.

The benefits of updating were also not clear to respondents, especially if the software was not perceived as important or if it was used rarely. P29 explains: *"Today I chose not to accept an update for iTunes. I generally do not update iTunes because I think there are too many of them and I don't regularly use the software."* [P29S1]

The costs of updating, however, were readily apparent, especially if a reboot was likely to be required or respondents were currently busy using the software. P130 describes needing to use the software:

> *Windows asks to update frequently. Because it is frustrating to be asked to update, or to say my computer must be restarted, when I am in the middle of work, I will put it off as long as possible.* [P130S1]

Updates could contain viruses

Respondents felt that the update itself could contain a virus and others were concerned about their ability to differentiate between real and fake notifications. Concerns caused some respondents to stop updating completely.

P61 read a news article about how the certificate used for Microsoft Updates was compromised:

> *Became very hesitant and did not install a Windows update, due to a virus that exploited the Windows update and allowed improper allows code signing to Microsoft certificate. This resulted in multiple "rootkit" virus that compromised the system.* [P61S1]

Concerns about viruses were only mentioned directly in seven stories but the implications are serious given that the concerns caused people to stop installing all updates.

Bundled software was also occasionally conflated with viruses. Bundled software is any third-party software added to the installation process that is not strictly necessary for the operation of the software being installed. It is sometimes included as sponsored content. A common example is Java which previously bundled the Ask Toolbar with its updates [**?**]. Some actions taken by the software such as changing the default search engine were seen as virus-type behaviors causing the respondent to determine that the update contained a virus. P141 talks about the viruses in Quicktime.

> *The update for Quicktime gave me problems because it comes with add on software. Some of the software has viruses and can harm the computer. It also has re-directs and takes over the search engines. I decided not to do updates from them unless I really need it.* [P141S2]

Some respondents also conflated the Ask Toolbar with malicious software, prompting P257 to use an antivirus to "clean" the computer. *"I had to uninstall it and run a virus check to make sure her computer was clean."* [P257S2] Bundled software is such a problem for consumers that Microsoft's Anti Virus program automatically uninstalls some versions of it, including the Ask Toolbar [**?**].

While mentioned in relatively few stories, it is concerning to see that respondents are starting to equate viruses with the updates intended to help protect them against viruses.

Preparation

Updates sometimes required preparation of the software or device before they could be safely installed. Preparation activities ranged from making sure a device had power, to creating a backup in case the update failed to meet expectations. This stage was not commonly mentioned in the data set, with only 29 stories explicitly mentioning a preparation type activity. Given the amount of power and data required by updates, we anticipate that many more people encountered this stage than mentioned it.

P268 sums up the issues around preparation in her story. Note that her goal is to update an app, which required an upgrade of the Operating System (OS), which in turn required other preparation activities.

> *I had a new app I wanted to install tell me it wasn't compatible with the version of iOS I had installed so I finally decided update from 8.2 to 8.3. First I'm told I haven't got enough space to perform the update, so I had to dig around and delete some apps I didn't want (it needed 1.6 GB on an 8GB phone, so it's not a small percentage required here!). Now I have to go back and reinstall those apps - annoying. Then it told me I needed more battery life in order to do the update so I had to go downstairs and plug the phone in.* [P268S1]

Backup first

Backing up files before an update was predominately mentioned by people upgrading the OS of their device. The action was typically taken as a safety behavior in case something went wrong. P216 talks about how he backed up in case he did not like the outcome of the update: *"While updating a Sony Playstation I noticed the new software update removed some of my abilities I had previously when I purchased the unit. I decided to back-up the current OS, and install the new update. Afterward I reverted because I felt like abilities I paid for were removed."* [P216S1]

Downloaded updates need disk space

Downloaded updates can be quite large relative to the amount of free drive space on devices, particularly phones and budget computers. To even download the update, respondents had to delete content such as music, photos, and apps off of their devices. P26 talks about managing limited drive space: *"I first had to delete half of the apps on my phone and also upload my pictures to my computer and delete them off of my phone because I didn't have enough usage to install the update."* [$P26S1$] The amount of drive space available for user content has long been a contentious issue for devices such as the iPhone and Surface [**?**, **?**]

Must be in a certain physical location to update

Physical location constraints also required some respondents to re-locate before they could update. One of the main causes of physical location constraints was data plans and limited network connections. P100 talked about being on a satellite uplink with limited data. P32, also using a satellite, explained how he had to travel to the library to get a free data connection, while P31 favored using the local McDonalds. However, not all devices can be easily physically relocated. P9 describes updating his XBox game platform: *"It was very difficult on the account that I didn't have home internet so I tried to use a friends computer to put the update on a flash drive."* [$P9S1$] While possible, updating using flash drives and trips to the library incurs costs for the user. P9's XBox update attempt failed causing him to abandon updating the device as too much work.

Installation

Once downloaded the update had to be installed. This process might involve some interaction from the user, computational resources, and rebooting. All of which required time during which the user might not be able to use the software.

The installer was the most talked about stage of the update process, with just under half of stories mentioning it. The most common comment about the installer was that it ran smoothly. Unsurprisingly, of the 103 stories which reported no installer issues, 70 were positive stories and 27 were neutral with only 6 negative stories mentioning no installer problems. Positive installation stories tended to be succinct and provide few details. P105 provided a typical story: *"[The update] went smoothly and I am happy with the improvements that it brought"* [$P105S2$]. The positive stories tended to use terms like "smoothly," "no problems," "intuitive," and "quick" to describe the experience. For other users, issues could occur at several points in the installation process.

Bundled software

Respondents rarely talked about the user interface of the installer with one notable exception: bundled software. Users can typically opt-out of installation of bundled software, but having to search for and un-check boxes on the interfaces makes the updating process more complex and risky. P207 describes how bundling software seemed sneaky. *"Two days ago my Java notice came up and reminded me to update. I decided to install the update. And, it is always tricky because they try to sneak the "Ask Toolbar" into the update. I opted no for the toolbar."* [$P207S1$] If installed, the bundled software was challenging and time consuming to remove. *"It was a time consuming hassle to reset my browser to its prior settings and to remove the unnecessary software which I had no use for."* [$P298S1$] The risk of problems motivated respondents to be careful when updating software and delay to when they had dedicated time.

Runtime resources

Resources were another contentious point for people during installation, especially if their device had limited CPU, RAM, or battery power. Budget devices, or devices already heavily laden with running programs, had limited resources to spare for the updating software. Respondents described having to close other programs, or not use the computer at all during update installation. Even supposedly background updates could cause a computer to inexplicably come to a crawl. P186 describes such a situation: *"Then [Windows Update] drains my processing resources, preventing back ups and even stopping me from running stats software for my job."* [$P186S2$]

Problems with resource contention sometimes led people to delay installing an update. P73 describes how anticipation of installer resource usage caused them to not update:

> *I decided not to update my Windows laptop when a bunch of new updates came out at the same time. I knew my laptop couldn't handle them all.* [$P73S2$]

Disruption

Unsurprisingly, the amount of time required to update was commonly mentioned. Negative and neutral stories were more likely to feature an installer that took what was perceived as a long time and positive stories more likely to discuss one that took a short time. In the follow-up survey for story1 respondents who installed the software were asked how much time the installer took, 43% of respondents though that the installer took "about as much time as I expected" and 45% thought that it took "more time than I expected."

While some applications can update in the background, many either require that the software be turned off or immediately restarted after the installation. This is done for practical reasons, since software can get into an unstable state if updated while running [**?**]. However, it also means that a user cannot reliably use software while it is updating, making the length of the installation a serious point of concern. Respondent described expecting a short installation period then being forced to remain in a particular physical location or loosing access to the device for longer than expected. P327 had both problems: *"I was sitting in my car ready to go to the golf course. I thought it would only take a few minutes to download the new info. It took over 20 minutes and while it was downloading I missed a call from my friend advising me that he wouldn't be able to golf that day."* [$P327S1$]

While some software only needed to restart itself causing minimal interruption to the respondent, other software required a reboot. When asked to relate a negative second story P152 stated that: *"The only time I think it's a negative experience is when I have to reboot my computer. I hate rebooting."* [$P152S2$] Her dislike of rebooting was common.

Troubleshooting

Respondents mentioned the need to troubleshoot at nearly every stage of the update process, but it was most commonly talked about in reference to handling failures in the installer or resolving issues with the post-state.

Troubleshooting the installer

A common issue with the installer was that it would fail for no visible reason, leaving the respondent with software that no longer functioned. When the installer failed respondents had two possible options: troubleshoot or revert to an earlier version of the software. P191 explained how his game software failed during installation causing him to try troubleshooting then reverting: *"The update would proceed for an hour and ultimately fail, preventing me from playing the game... Tried again and it occurred again. Ultimately had to uninstall the software and reinstall it from scratch."* [P191S2]

The most straight-forward solution for most respondents was to revert either by reinstalling the software as P191 described about above, or by using Windows' system restore functionality which allows a user to roll back the system to right before the installation of any large piece of software. P291's story was typical:

> *My Advast Virus software wanted to update to a new version of the software. I completed this update, however then my computer gave me the Blue Screen of Death. So, I went back to a restore point before the update. My computer has been working fine ever since, but I have still not updated the program.* [P261S1]

While Windows provides a mechanism to revert software, not all devices did, forcing users to troubleshoot. P293 describes how an update slowed his iPhone to "a crawl" so he tried to fix it. When his troubleshooting efforts were unsuccessful he tried to get professional help fixing it:

> *I took it to the Apple store had to wait in line for 45 minutes and then they refused to help me unless I paid a huge fee and even then they couldn't guarantee my phone would be fixed. They tried to sell me a new phone and I told them I wasn't interested ....* [P293S1]

What was perhaps the most remarkable was the number of respondents who persisted in getting their software updated despite installer failures. 58 stories described situations where the respondent encountered problems with the installer, yet only 10 describe a situation where the problem was considered insurmountable and the respondent gave up trying to fix it. This finding is likely caused by our sampling bias. Our sample is more technically skilled than the average internet user which might explain their willingness to troubleshoot installer problems.

Troubleshooting after install

Even after a successful installation, troubleshooting was still required for some respondents who had to put effort into returning elements of the software to their pre-update state. Settings had to be checked and sometimes restored. P77 describes fixing her settings after updating. *"I recently updated my computer from OS X Snow Leopard to OS X Yosemite ... requiring me to reconfigure a lot of settings to my preferences."* [P77S1] P64 had a similar problem: *"Recently I installed an iOS update for "bug fixes" though it got rid of my shortcut key configuration. Pretty bummed about it."* [P64S1] Initial set-up work sometimes also had to be done again. P268 describes an iOS upgrade: *"once installed, Apple treats it almost like a new phone, you have to relogin, accept decline cloud services and agree to their 800 page (feels like it) terms and conditions which no one in their right mind ever reads."* [P268S1]

Updating the software might also cause data to be lost either due to voluntary deletion during the preparation stage, or due to an error with the update. P231 describes using a backup to restore: *"I had to restore my data from my most recent backup and try to install the update again through iTunes. It was a frustrating experience."* [P231S2] This loss was an annoyance for respondents who had remote backup as they had to spend time restoring the data over a network connection. P182 explains: *"I updated my steam gaming platform and lost every game I had on it. I was pretty mad because my internet is slow and it took forever to download the games again."* [P182S2] However, not all respondents were lucky enough to have backups. P294 describes learning about backup failure only after an update wiped out his data: *"When I finally got it to install the new update wiped out all my data and I found that my phone had not been backing up properly. This was an incredibly frustrating experience because I keep a lot of information for my job on my phone. It took a lot of will power to not throw my phone across the room."* [P294S1]

Post state

When describing the post state of the software, respondents who had a positive experience tended to talk about how there were no problems, more features, and better performance. Respondents who had a negative experiences tended to talk about how they had more problems, worse performance, and how other software was also worse.

Similar to other stages, respondents who had a positive experience tended to be terse when describing them. P27's story was typical: *"Google drive wasn't working, I updated and that fixed the problem."* [P27S2] Respondents, particularly when asked to describe a positive second story, tended to talk about how an update fixed a problem that they had been having. P252 described how an update fixed problems with her software. *"One of my programs would not work properly. I'd been delaying a recommended update so I finally updated and, not surprisingly, the program began to function correctly again."* [P252S2] Respondents talked about how the update improved specific functionality, made the software run "smoother", solved "problems", and corrected "bugs". Functionality improvements were diverse ranging from applications that now allowed photos to be taken to games that added new levels.

Bugs and lack of stability in the new version were a source of annoyance and caused some people to consider reverting the software. P113, who updated right away, best described this reaction to the unanticipated number of problems.

> *I installed iOS 8.0 the day it came out. I was shocked at the amount of bugs that occurred within the OS and that it had made it to Gold Master in his state. I wanted to revert back until they figured it out it was so bad. Apps froze, landscape mode got stuck and many other annoying this as well.* [P113S1]

Compatibility was another issue, particularly with automatic updating where one software might update, but another dependent piece of software did not. P117 described how an automatic update caused her to loose important functionality: *"The computer automatically updated a version of Flash Player which disabled my access to sites I really needed for work. I had to uninstall it and it took a great deal of time. This was frustrating."* [P117S2]

DISCUSSION AND CONCLUSIONS

When interacting with updates, users are balancing the risks and costs of updating against potential benefits. This work details the process users go through when updating software as well as the types of issues respondents encountered during each stage that impacted their willingness to update. Below we discuss some of the implications this work has for designers and software developers.

Make it easy to find information about an update.

Users want to know what installing an update will involve and if they are likely to enjoy the changes it will make. We observe that respondents put effort into researching updates, asking friends and family, as well as reading online comments. The issues they were most concerned about included: time required to install; resource usage during installation as well as after installation; if the update improved the software; and if it impacted features they valued. App stores such as Google Play are making it easier to find this type of information by providing a place where users can read reviews from other users. Google Play also provides a "What's New" section where developers can communicate what an update contains. However, developers currently have minimal guidance about what to write about the update and users still have to dig through comments for important information.

Be conscientious of resources.

Considerable research effort has gone into exploring how to efficiently propagate updates across various types of networks [**?**, **?**], yet very little work has looked at how to install those updates on resource constrained devices. Respondents had issues with resources such as disk, processor, and data throughout the update process. For example, delaying an update due to lack of disk space or a limited data plan was an issue for iOS users on space constrained devices. Update developers should be conscientious of the limited availability of these resources on devices and provide options that allow users to update while managing their resource constraints.

Provide a recovery path for users.

Installing an update is risky for users, they may not like the new features, the update may fail, it may introduce new bugs, or make their computer slower. The possibility of failure without the certainty of a recovery strategy makes updating more risky for users . One of the most common approaches our respondents used to handle bad outcomes was to revert their software to an earlier state. This was well supported on Windows 7 and 8 for full applications where the Operating System automatically makes a restore point before large changes. Respondents described using this feature to revert back to a restore point if they disliked an update. Similarly, Apple provides automatic backup functionality on some of its devices which enables a user to recover their files in the case where an update damages them. However, this functionality is not provided on all devices and focuses on supporting file recovery over applications making it harder to revert to an earlier version of an application without reverting the whole computer. Reverting apps on mobile devices is also poorly supported.

Willingness to use backup recovery and reverting Windows may also be a side effect of our overly technical population. We anticipate that using Windows' restore point is challenging for an average user and a survey of the general population would find that the feature is not a practical choice for most people. We therefore recommend that designers and software developers create a clear path for people who want to revert their software to an earlier version.

LIMITATIONS

When designing the survey we were primarily interested in what aspects of updates were most salient and memorable to users. This goal caused us to ask respondents to tell us about a memorable update experience rather than the most recent or most frequent one. Therefore, this work does *not* represent the frequency of the different update events discussed, merely that they do happen to respondents. Due to this limitation we only provide count information when the number of stories mentioning the topic is particularly high or low to highlight update aspects which many people recalled, or were rare.

We used Amazon's Mechanical Turk [**?**] to recruit respondents. The demographics of Mechanical Turk differ from the general United States population in that they typically have more technical experience and are more privacy conscious [**?**]. In our demographics questions we asked participants if they ask others for help, and if others ask them for help [**?**] to gage their computer experience self efficacy. We found that indeed our population is more confident in their technical ability, rarely asking others for help and often receiving requests for assistance.

The oversampling of technically literate respondents is an important limitation of this work. This demographic is potentially more likely to install a broader range of applications, they may also spend more time troubleshooting problems when updates do not work as expected, and they may be more willing research updates. However, we argue that this demographic is still a very important one to study. As Poole et al. observe in their work on informal technical support, these local experts often provide technical assistance to the people around them [**?**]. Their experiences and opinions are likely to be shared with others impacting a much larger set of users through advice and stories [**?**].

REFERENCES

1. 2015. Amazon Mechanical Turk. (2015). Accessed Sept. 25, 2015 https://requester.mturk.com/.
2. Hazim Almuhimedi, Adrienne Porter Felt, Robert W. Reeder, and Sunny Consolvo. 2014. Your Reputation Precedes You: History, Reputation, and the Chrome Malware Warning. In *Symposium on Usable Privacy and Security*.
3. Leyla Bilge and Tudor Dumitras. 2012. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security (CCS '12)*. ACM, New York, NY, USA, 833–844.
4. Cristian Bravo-Lillo, Lorrie Cranor, Saranga Komanduri, Stuart Schechter, and Manya Sleeper. 2014. Harder to Ignore?. In *Symposium on Usable Privacy and Security (SOUPS)*.
5. Cristian Bravo-Lillo, Lorrie Faith Cranor, Julie Downs, Saranga Komanduri, and Manya Sleeper. 2011b. Improving computer security dialogs. In *Human-Computer Interaction (Interact)*.
6. Cristian Bravo-Lillo, Lorrie Faith Cranor, Julie S Downs, and Saranga Komanduri. 2011a. Bridging the gap in computer security warnings: A mental model approach. *Security & Privacy, IEEE* 9, 2 (2011), 18–26.
7. Cristian Bravo-Lillo, Saranga Komanduri, Lorrie Faith Cranor, Robert W Reeder, Manya Sleeper, Julie Downs, and Stuart Schechter. 2013. Your attention please: desgning security-decision UIs to make genuine rrisk harder to ignore. In *Symposium on Usable Privacy and Security*.
8. Peter Bright. 2015. Microsoft accused of adding Windows 10s spy features to Windows 7 and 8. Web. (Sept. 2015). http://arstechnica.co.uk/information-technology/2015/09/microsoft-accused-of-adding-spy-features-to-windows-7-8/
9. Jean Camp, Tim Kelley, and Prashanth Rajivan. *Instrument for Measuring Computing and Security Expertise–TR715*. Technical Report.
10. L Camp. 2003. Designing for trust. *Trust, Reputation, and Security: Theories and Practice* (2003), 203–209.
11. CNN Political Unit. 2013. McCain: 'Why the hell' do iPhone apps need updating? (May 2013). http://politicalticker.blogs.cnn.com/2013/05/21/mccain-why-the-hell-do-iphone-apps-need-updating/.
12. Lorrie Faith Cranor. 2008. A Framework for Reasoning About the Human in the Loop. *UPSEC* 8 (2008), 1–15.
13. Tudor Dumitras, Priya Narasimhan, and Eli Tilevich. 2010. To upgrade or not to upgrade: impact of online upgrades across multiple administrative domains. In *Proceedings of the ACM international conference on Object oriented programming systems languages and applications (OOPSLA '10)*. ACM, New York, NY, USA, 865–876.
14. W.K. Edwards, E.S. Poole, and J. Stoll. 2007. Security automation considered harmful. In *Proceedings of the IEEE New Security Paradigms Workshop (NSPW 2007)*. 18–21.
15. Michael Fagan, Mohammad Maifi Hasan Khan, and Ross Buck. 2015. A study of users experiences and beliefs about software update messages. *Computers in Human Behavior* 51 (2015), 504–519.
16. Samuel Gibbs. 2015. Apple faces lawsuit over storage space on iPhones and iPads. (2015). http://www.theguardian.com/technology/2015/jan/02/apple-lawsuit-storage-space-iphones-ipad-ios8-software-advertised-capacity.
17. Christos Gkantsidis, Thomas Karagiannis, Milan VojnoviC, Christos Gkantsidis, Thomas Karagiannis, and Milan VojnoviC. 2006. Planet scale software updates. In *ACM SIGCOMM Computer Communication Review*. ACM, New York, New York, USA, 423–434.
18. Dan Goodin. 2015. Ding dong, the witch is dead: Microsoft AV gets tough on Ask Toolbar. ArsTechnica news site, Accessed Sept. 12 2015. (June 2015).
19. Bela Gor and David Aspinall. 2015. Accessible Banking: Experiences and Future Directions. In *Workshop on Inclusive Privacy and Security (WIPS)*.
20. Iulia Ion, Rob Reeder, and Sunny Consolvo. 2015. ...no one can hack my mind: Comparing Expert and Non-Expert Security Practices. In *SOUPS15*.
21. Ruogu Kang, Stephanie Brown, Laura Dabbish, and Sara Kiesler. 2014. Privacy Attitudes of Mechanical Turk Workers and the U.S. Public. In *Symposium On Usable Privacy and Security (SOUPS '14)*. 37–49.
22. Dara Kerr. 2012. Microsoft hit with lawsuit over Surface's storage space. (2012). http://www.cnet.com/news/microsoft-hit-with-lawsuit-over-surfaces-storage-space/.
23. M. Khan, Zehui Bi, and J.A. Copeland. 2012. Software updates as a security metric: Passive identification of update trends and effect on machine infection. In *Military Communications Conference (MILCOM)*. 1–6.
24. Kat Krol, Matthew Moroz, and M. Angela Sasse. 2012. Don't work. Can't work? Why it's time to rethink security warnings. In *Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on*. 1 –8.
25. Muhammad Mahmoud, Sonia Chiasson, and Ashraf Matrawy. 2012. Does Context Influence Responses to Firewall Warnings?. In *eCrime Researchers Summit (eCrime)*.
26. G.V. Marconato, V. Nicomette, and M. Kaaniche. 2012. Security-related vulnerability life cycle analysis. In *Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on*. 1–8.

27. Eduardo R. B. Marques. 2013. Fine-grained Patches for Java Software Upgrades. In *5th Workshop on Hot Topics in Software Upgrades*.

28. Microsoft. 2012. Microsoft Security Intelligence Report, Volumne 13. (January – June 2012).

29. Matthew B Miles, A Michael Huberman, and Johnny Saldaña. 2014. *Qualitative Data Analysis: A Methods Source Book*. SAGE Publications, Incorporated.

30. David Moore, Colleen Shannon, and k claffy. 2002. Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment (IMW '02)*. ACM, New York, NY, USA, 273–284.

31. Antonio Nappa, Richard Johnson, Leyla Bilge, Juan Caballero, and Tudor Dumitras. 2015. The Attack of the Clones: A Study of the Impact of Shared Code on Vulnerability Patching. In *Symposium on Security and Privacy*.

32. Kartik Nayak, Daniel Marino, Petros Efstathopoulos, and Tudor Dumitras. 2014. Some Vulnerabilities Are Different Than Others: Studying Vulnerabilities and Attack Surfaces in the Wild. In *International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*.

33. Jared Newman. 2015. Java installer ditches the Ask Toolbar, swaps in Yahoo defaults. Online, Accessed Sept. 12 2015. (June 2015).

34. Srivatsan Parthesarathy, Reiner Fink, Sean L Flynn, and Ray Sun. 2002. Software update notification. (March 5 2002). US Patent 6,353,926.

35. Erika Shehan Poole, Marshini Chetty, Tom Morgan, Rebecca E. Grinter, and W. Keith Edwards. 2009. Computer help at home: methods and motivations for informal technical support. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*. ACM, New York, NY, USA, 739–748.

36. Emilee Rader, Rick Wash, and Brandon Brooks. 2012. Stories as informal lessons about security. In *SOUPS '12: Proceedings of the Eighth Symposium on Usable Privacy and Security*. ACM.

37. Eric Rescorla. 2003. Security holes... Who cares?. In *USENIX Security*. Washington, DC.

38. Johnny Saldana. 2013. *The Coding Manual for Qualitative Researchers*. SAGE Publications Ltd.

39. David Sharek, Cameron Swofford, and ichael Wogalter. 2008. Failure to recognize fake Internet popup warning messages. In *the human factors and ergonomics society*. Study whre showed real and fake Windows XP Warning windows. No difference between real and fake.

40. Andreas Sotirakopoulos, Kirstie Hawkey, and Konstantin Beznosov. 2011. On the Challenges in Usable Security Lab Studies: Lessons Learned from Replicating a Study on SSL Warnings. In *SOUPS11*.

41. Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. 2009. Crying Wolf: An Empirical Study of SSL Warning Effectiveness. In *The 18th USENIX Security Symposium*.

42. Symantec Corporation. 2014. Internet Security Threat Report, Volume 19. (2014).

43. Yuan Tian, Bin Liu, Weisi Dai, Blase Ur, Patrick Tague, and Lorrie Faith Cranor. 2015. Supporting Privacy-Conscious App Update Decisions with User Reviews. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '15)*. ACM, New York, NY, USA, 51–61. DOI: http://dx.doi.org/10.1145/2808117.2808124

44. Peleus Uhley. 2013. Strategies for Updating at Scale at Adobe. Keynote address at 5th workshop on hot topics in Software Upgrades. (June 2013). https://www.usenix.org/conference/hotswup13/workshop-program/presentation/uhley.

45. Kami Vaniea, Emilee Rader, and Rick Wash. 2014. Betrayed by Updates: How Negative Experience Affect Future Security. In *CHI 2014: Conference on Human Factors in Computing Systems*.

46. Rick Wash. 2010. Folk Models of Home Computer Security. In *Proceedings of SOUPS*.

47. Rick Wash, Emilee Rader, Kami Vaniea, and Michelle Rizor. 2014. Out of the Loop: How Automated Software Updates Cause Unintended Security Consequences. In *Symposium on Usable Privacy and Security (SOUPS)*.

48. Leah Zhang-Kennedy, Sonia Chiasson, and Robert Biddle. 2014. Stop Clicking on "Update Later": Persuading Users They Need Up-to-Date Antivirus Protection. In *Persuasive Technology*.