Abstract
This paper concerns the cultural phenomenon of software update unrest. It focuses on two recent cases of update unrest surrounding Tinder and Tesla that likewise highlight the growing centrality of digital social media for negotiating power dynamics between software developers and end users. We discuss the similarities and differences between the two cases and address how they compare with the well-known iOS6 Maps update unrest from 2012. We also analyze the specific role played by DSM and highlight some of the new types of DSM user behavior and communities that recently emerged around Tinder and Tesla update unrest activities. At stake in studying software update unrest, in our view, is not just matters of aesthetics and habits of use, rollout blunders and customer relations, but also, and more importantly, remotely enacted, highly social, and heavily mediated debates over how technology is made.

1. Introduction

Digital social media (DSM), like other software technology, is made through updates. Updates are a way of making and remaking software technology “in the wild,” in real time, over sometimes long periods of time. This temporally distributed way of making and remaking software technology means that software development processes do not necessarily result in “final” products, tools, or things. It also means that software technology user practices, including DSM user practices, have the potential to be continuously upended in a manner distinct from the user practices and “stabilization” [1] processes associated with older forms of technology, like bicycles or missiles for instance.

There is now widespread update unrest on the part of software technology users and DSM has a gained a prominent role within update unrest culture. By “unrest” we mean public demonstration, protest, online dissent and cross-platform discourse that exceeds (in its content, complexity and potential to influence) mere “dissatisfaction” on the part of software consumers. DSM routinely serves as a venue for organizing update unrest against other software technologies (e.g., operating systems, databases, etc.) and DSM is also now a common impetus for update unrest when updates are made to popular social media platforms. Software technology is increasingly made through a social churning: back and forth flows of making and using that contain a considerable amount of turbulence. Whether similar types of public unrest would exist if software was made in other ways is impossible to know. What makes update unrest all the more fascinating, though, is that software updates (much like DSM) are highly participatory in their animating logics [2]. In addition to reflecting cross-domain advancements in technical knowledge, shifting design sensibilities, and changing market trends, many updates are direct responses to user feedback and experience—a sincere and collaborative effort on the part of information, computer, and system scientists to improve, enhance, fix, or advance software technology already-in-circulation according to the needs and demands of ordinary users. Updates are also participative in at least one other way: they typically involve end users in the update installation process, an arrangement that dates back to the era when software updates came exclusively on physical media (e.g., floppy disk, CD-ROM) and required what at times could be lengthy and tedious circuits of non-networked transmission involving everyone from plastics manufacturers to truck drivers and postal workers.

Given these participatory elements within software updating practices, why does update unrest flourish as individual and collective acts of challenge to fairly established technology production practices? And what role does DSM play in negotiating power dynamics between software developers and end users
engaged in update unrest activities? As a way to begin answering these questions, and in the hopes of thickening the thinking around both software updates and DSM, this paper, a cross-platform study, presents the results of research into two cases of recent update unrest that occurred during the past 365 days. Each case involves DSM in slightly different ways. The first case concerns users of Tinder, a popular dating application that in 2015 revamped user profiles by more strongly integrating data from ancillary social media platforms and limiting the number of matches users can make within specific windows of time—leading to widespread update unrest on app review platforms. The second case concerns a series of updates made in the fall of 2014 by Tesla Motors, Incorporated that were perceived by many Tesla owners to be unfairly implemented. In response to the updates, Tesla owners organized update unrest using the popular Change.org platform. In each case, DSM proved central to coordinating update unrest activities, but in each case differences between DSM platforms also shaped the contours of the update unrest in question.

In carrying out this study, we approached DSM as a diverse but coherent category of software technology. But we also took as our starting point that DSM, like much other software technology, is “technology-in-the-making,” to borrow a term from Nelly Oudshoorn [3]. In other words, the DSM increasingly entangled with update unrest activities is itself consistently changing, as well as diversifying. Rather than a problem or shortcoming, this muddy and mutable situation is what makes studying update unrest and its cross-traffic with DSM so fascinating. We also adopted as our starting point that updates themselves vary in nature. Most updates fall somewhere between two poles: the refresh update and the rebuild update. The refresh update involves serving up something intuitively the same to the user: the interface and underlying information systems somehow look and feel fresh however the backend system for designers and administrators is relatively unchanged. At the other end of the continuum is the “rebuild” update, which comes from the language of mechanics (for example, “rebuilding an engine”). The rebuild update often involves starting over from the ground up, building from scratch, and re-engineering the user experience anew. In contrast to refresh updates, rebuild updates usually involve large, sweeping changes in design, features, and permissions (i.e., how the data that is generated by the user is collected and utilized by the developer). The grammar of the information system is substantially changed and users can feel these changes in dramatic ways. In addition to refresh and rebuild updates, many updates are for bug and security fixes. Sometimes these patches represent updates within updates, after larger versions have been released and bugs have been identified and reported by users. There are also updates designed to ratchet down products that no longer represent a viable area of development and support.

This variety among update motivations and practices means that update unrest amongst software users likewise varies. Perhaps the most famous case of update unrest in recent history involves Apple, Incorporated. In September of 2012, Apple released the iOS6 mobile operating system update and 60% of iPhone users updated platforms on their mobile devices from iOS5 to iOS6 [4]. The platform update included the replacement of the Google Maps mobile application in exchange for a new, vector-based engine developed by Apple called “Maps.” As a result of the iOS6 Maps (rebuild) update that replaced Google Maps on the iPhone, 150 million iOS users instantly lost public transit navigation capabilities with the Maps app and intense public criticism ensued. Within a few hours of the iOS6 update’s rollout, Apple iPhone users turned to DSM platforms—from Tumblr to MacRumors.com—to voice criticism and debate about the lack of accurate public transportation data (e.g., train and bus arrivals and departures). Users also began to publicly catalogue direction and location errors that involved basic navigation inconsistencies and mismatches between information on Maps and offline transportation infrastructure. For months afterwards the Maps app update was parodied on popular technology blogs and on television programs.

Update unrest, as we aim to demonstrate, is in fact a much broader cultural phenomenon that spills well beyond the famous case of iOS6. Yet update unrest remains under-documented and under-theorized, along with the critical role of DSM within update unrest culture. An insufficient amount of “active listening” [5] is happening when it comes to update unrest in, on, and around DSM, where different actor groups are engaged in potentially meaningful disputations about not just surface features and functionalities but also about how technology is made. Moreover, leaving update unrest under-studied risks conjuring software technology users as inherently and consistently conservative—as social actors universally rooted to what might be called the “long then” of past and current software versions.

As a means to begin filling this critical gap in the existing literature, patching a hole, this paper unfolds in the following manner: in the next two sections we provide an overview of the recent update unrest surrounding Tinder and Tesla. We then discuss the
similarities and differences between the two cases, and address how they compare with the well-known iOS6 update unrest that took place in 2012. We also analyze the specific role played by DSM and highlight some of the new types of DSM user behavior and communities that emerged around Tinder and Tesla over the past year in response to software changes. Our larger argument is that update unrest is evidence that users are coming to terms with software development in new and interesting ways, including living with and relating to updates in more interactive, disputatious ways.

2. “Your Revamped Tinder Profile”: Unrest around trans-platform integration

Tinder is a location-based dating app that allows users to swipe through profiles of other nearby users with a few preferences (gender, age range, and location). It was released for iOS and Android users in 2012 as a smartphone app. Users create a simple Tinder profile by connecting their Facebook account through app permissions between the platforms. User names, ages, and Facebook social graph information are used by Tinder to provide matches. With this Facebook integration, users can also select between 1 and 6 photographs from Facebook albums to feature on their Tinder profile. Users write a short description below their profile known as a “bio” and select a few preferences for the kinds of matches they would like to swipe through. A potential match based on your preferences and location is presented to a user and they can swipe left (“NOPE”) or swipe right (“LIKE”). If there is a mismatch between users, that is, if one swipes right and another user swipes left, each user’s profile is hidden from the two streams. Unmatched profiles never appear in either user’s stream again. A “match” occurs when two Tinder users both swipe right (hereafter, “swipe-like”) each other. Users that swipe-like each other are notified in the app that a match has occurred and they subsequently appear in a matched queue that allows the sending of messages back and forth. Users can also send “moments” or ephemeral picture messages to matches as part of the messaging function [6].

In the spring of 2015, Tinder updated to version 4.4 announcing to users through their blog, “We Just Revamped Your Tinder Profile” [7]. The 4.4 update offered stronger Facebook integration for better “Common Connections,” Instagram integration, and a limit on swipe-likes within a 12-hour time period. Within a few hours, hundreds of users began to complain of their revamped profiles and matching experiences on DSM platforms like Twitter and on the app reviews and feedback sections of the Google Play Store and iTunes App Store, which distribute mobile apps to Android and iOS devices [8]–[10].

A few things make Tinder different than other online dating platforms such as OkCupid or Match.com [11], [12]. First, Tinder is strictly a mobile app; it uses pictures, permissions, and social graph data from your Facebook profile. That is, you have to be a member of Facebook in order to use the Tinder app [12]. Second, older, data-driven platforms like OkCupid, eHarmony or CoffeeMeetsBagel are based on matchmaking algorithms where users provide lots of information about likes, dislikes, interests, and hobbies to facilitate the work of filtering algorithms that aim to produce potential matches [13]. In contrast, Tinder limits their matching algorithm to the 3 preference selections (age, gender, and location); privileges the current location data of the user’s phone; and relies upon users to quickly choose matches up to 100 miles away by swiping through a stream. In order to look at the next profile you have to swipe on the most current profile in your potential match stream. Tinder has been praised for its simplicity, especially the low barriers to getting started and matched with prospective users. It has also been touted as an online solution for women in particular, because Facebook integration offers a kind of “social accountability” for vetting people that use the world’s leading social network site with a real name policy [13]. But like many online dating platforms, e.g., Grindr, Tinder is filled with fake user accounts known as bots who engage with mostly male-identified accounts as simulated female users willing or wanting to chat.

The Tinder 4.4 revamp update incorporated more social accountability from neighboring platforms by marshaling more social graph data from user’s Facebook profiles (friends of friends, and interests based on likes) and allowed users to integrate their Instagram profiles directly into their Tinder profiles (up to 35 photographs). Before Tinder 4.4, many user bios simply contained Instagram (IG) usernames, a strategy that Tinder developers identified and addressed with the revamp’s “slick” rollout. The updated Common Connections feature aimed to help users find matches with mutual Facebook friends in common, and friends of friends. When the update first rolled out on April 15, 2015 many users experienced down time and service disconnections, reporting them to a number of DSM platforms including Facebook, Twitter, Google Play Store, iTunes App Store, and app rating sites [14]. User reports from DowntimeDetector posted on April 15, 2015 show that most of the outage reports were from server connections (43%), and of complaints about
sending and receiving messages (32%) [15]. Tinder announced the service outages to users in a tweet, “GOOD NEWS: Everyone seems to be checking out their revamped Tinder profile. BAD NEWS: Tinder might be slow / down in the next 24h [monkey emoji]” [8].

Update unrest amongst Tinder users took a few forms in the coming days and weeks after the revamp. First, users took to reviewing the update in the Google Play store and the Apple iTunes store, reporting outages, loss of messages, and old match lists, including chats. Some users, like Amber, alerted folks on Twitter to the unrest on the app stores, tweeting, “Read Tinder AppStore reviews for a good time” screen grabbing a complaint from a user about a match list being empty [16]. Users began to sound off complaints about the revamp on DSM sites, likening bugs in the rollout to critical infrastructure disasters for disconnecting them from earlier matches and chat logs [8], [17]. Users who found their experiences echoed by other frustrated updaters, reported disappointment and warned prospective users not to download or agree to the forced update. Some threatened to uninstall the app or stop using it until the update itself was updated. For example, David, who wrote his one-star Google Play store Tinder review just hours after the update was rolled out, said: “I dont[sic] understand for an app with so much notoriety. The problems stated in all these comments and nothing has been fixed in weeks. If i see a change in the developers[sic] behaviors i will rerate but until then it remains. What good is creating something and not maintaining it” [18].

At the same time, online media outlets began reporting that the new Common Connections integration with Facebook and the likes lists in particular were “ruining” the experience of the app for users [17], [19], [20]. These new integration features included much more data in user profiles, and many tech writers noted that such features were bothersome (i.e., provided too much information for some users to have fun swiping through streams quickly). Journalists forecasted pushback from users about stronger and more integrated profiles based on social graph data. Reports about the update pulling Facebook likes [21] to appear as common interests in Tinder profiles appeared cumbersome, and possibly embarrassing, particularly for longtime Facebook users who have been liking stuff (ranging from toiletries to shoes and wine labels) for close to ten years. The social accountability that the limited social graph data afforded before the update was now dialed up to a degree that included things such as books, movies, and outdoor hobbies. Minutia such as Amazon purchases or Netflix recommendations was taking away from the simplicity of the experience design, according to users. Other users and journalists lamented that crowding profile streams with Instagram photos added another layer of self-presentation to the platform that was unneeded, and would certainly curb swipe-likes with more to choose from—by going from 6 photos to “1,000 IG photos [you] just made your impeccably-curated Tinder profile a rambling, long-winded mess” [22]. While the media reported that the stronger cross-platform integration would crowd or ‘weigh down’ the profiles with too much information, most users commenting in the app store reviews worried about unknown (and unmatched) people getting access to their personal IG account, and then having to deal with unwanted interactions with (not yet matched) Tinder users on another DSM platform [23].

Another form of unrest that occurred around the 2015 Tinder update involved users expressing frustration in the update not having fixed the problem of spambots on the platform [24]. For example, a four star review on the iTunes App Store cautioned: “Cant’ [sic] speak to the issues women have, but for guys the ‘sex bots’ are getting too good! Previously they would have two pics, no bye [sic] and cut to the chase. Now, they have an indistinguishable profile and will have a normal convo with you” [23]. The 4.4 update had indeed tried to address this problem by containing a new swipe-limiting feature. Tinder users themselves pointed to the responsibilities of the software developers to address the problem of bots through frequent software updates. Most users suggested that new updates should be aimed at users filtering out chat bots themselves by reporting them to the developers. Yankees13.24 wrote: “Would be a fantastic app but it has way to[sic] many, what I call internet prostitutes! These spam bots have even gotten so smart that they are even putting fake info in the bio part of these accounts to lure you in to swiping right![…]It’s a crap app that really needs to be filtered through by the developers and all the spam bots need to be kicked off after the first report file against them” [25]. Symantec security researchers [26] found that spamming campaigns on social media sites involve spambots that are programmed to flirt and solicit a users phone number or to encourage them to visit subscription based dating and escort service sites that promise nude pictures, porn, and webcam conversations. Popular media outlets like Huffington Post and Vice have offered tips on how to identify and avoid spambots, which have evolved from using explicit language to more subtle, normal “girl next door” conversation styles [26]–[28]. With the 4.4 update, Tinder now addresses both the bots and real users (who we refer to as “opportunistic
optimists”) that behave similarly to programmed bots by limiting their ability to swipe-like over a certain period of time. In the short-term, the update limit addressed bots that are programed to always swipe-like profiles, the problem is that some users always swipe-right, too. Tinder explained the limit feature algorithm and its motivations to address a particular kind of user behavior and to encourage more thoughtful use: “[T]o encourage people who right swipe like crazy to be thoughtful in their swiping, we’ve introduced an algorithm that intelligently limits the number of likes a user can make in a consecutive 12 hour period” [29]. What is interesting about the limiting feature of the revamp update, however, is that in its attempts to address a problem for some users (that is, getting matched with fake profiles) it necessarily limits other kinds of users who may use the app in similar ways as bots. As a result, opportunistic optimistic Tinder users who “swipe like crazy” and are interested in swipe-liking a lot of people in more than a 12 hour period in their stream are rendered as bot-like themselves, and are curbed from using the app in particular ways.

For Tinder users, the unrest against their revamped profiles took different forms, from reporting initial service outages and chat log loss to pushback against intra-platform integration with Facebook and Instagram as part of the new, expanded profiles that upended the simplicity of earlier match profiles with more social graph data (which “borked” the experience of the app by crossing too many DSM platforms together). For other users, the update’s swipe-like limiting feature still does not solve the problem of getting spammed by fake profiles as the bots have adapted their swipe “crazy” tactics and have changed their messaging language. Still for other users, limiting the amount of swipe-likes and the time period for which you can use the app represents an oppressive and universalistic way of encouraging particular patterns of use. The 4.4 update unrest from Tinder users shows that there was a disconnect between expectations of use between users and developers: by building in more trans-platform integration, the match streams become “crowded” with more social graph information, foregoing earlier, “simpler” profiles that promoted the app’s ease of use. The disconnect between expectations of use can also be seen in the limiting feature that guards against particular styles of use, now restricted (and left behind) with the new update.

3. Early Adopters Feeling Left Out: The 2014 Update Unrest Surrounding Tesla

In the autumn of 2014, Tesla announced a series of updates to its Model S that, in the eyes of many Tesla owners, seemed to be unfairly rolled out. The S was first introduced in 2012 and an enthusiastic buyer base had emerged by 2014—in part because a Tesla can be conceptualized (and marketed) as “a very sophisticated computer on wheels” [30]. The updates involved software for existing Model S vehicles, those already-in-circulation, but also hardware updates at the factory for all new Model S vehicles. The latter of which (i.e., the hardware updates) could not be done as a retrofit to vehicles already-in-circulation and yet would be necessary to receive and make use of future, related software updates. More specifically, in September 2014 Tesla announced an over-the-air software update called Software v6.0 that would add new features and functions to existing vehicles such as real-time traffic navigation, calendar modifications, new energy saving options, and more. The company framed the updates in the following manner: “This software update represents Tesla’s commitment to improving Model S for customers even long after it has left the assembly line. With each update, Model S becomes more attuned and responsive to its owner’s needs without requiring excessive user input. We will continuously fine-tune the software and work on new features in response to customer feedback” [26]. Shortly thereafter, in October 2014, Tesla then introduced a series of hardware updates for all new S vehicles that included parking sensors, fog lights, “autopilot” capabilities, and more— features that would be further updated and potentially enhanced over time with additional software, as it became available. New owners who purchased and received their vehicles in the weeks and days leading up to the updates were not informed that such updates were on the horizon and received vehicles lacking the updates, and lacking the ability to participate in certain future updates.

In response to these developments, a group of Tesla owners quickly began to organize on Change.org, a DSM platform that allows users to create petitions as well as craft and share explanatory messages about why someone supports a particular petition. Since its launch in 2007, Change.org has given rise to new modes of consumer activism, environmental activism, political lobbying, food politics, and the like. The Change.org petition created around the Tesla updates [31], which grew to involve nearly 1000 participants from North America, Europe, and Asia, was officially framed as a petition about the “autopilot” features but participants ultimately used Change.org to stage a far broader discussion about Tesla ownership, update practices,
and technological change over time—in addition to a discussion about the obvious problem that their vehicles had just lost considerable resale value due to their lack of updates.

Of particular concern within the update unrest was that many Tesla owners perceived themselves to be “early adopters” and thus a uniquely important stakeholder group. For Tesla to leave the updates unannounced, and to allow people to purchase the luxury vehicles in the weeks and days leading up to the updates, appeared to many participants as not only non-transparent but also to penalize “early adopters” by unfairly privileging new and future owners. The following type of statement is representative of how this point came to be expressed within the update unrest: “Tesla has always stated that the car would be upgraded continuously so that it always would stay updated and current...Tesla is blocking off the possibility for early adopters and supporters to take advantage of a huge part of the future upgrades” [32]. Said another owner: “We were told, that this car will be updated in the future with new functions! But after one year there is no way to retrofit important features as road sign recognition or autopilot. As an early adopter I feel punished and frustrated” [31].

Another, related point of concern that came to be frequently articulated within the update unrest was that Tesla owners believed their cars were already capable of all updates; confusion came to be expressed by the Tesla owners as to whether or not the vehicles were truly updateable as marketed. According to one owner: “My hopes for bying [sic] this car was that I did not have to change car that often due to changes in new models, this was the main point of bying [sic] the Tesla since it was promised [sic] to be a better car all the time by updates” [33]. Added another owner: “[W]e are part of a movement and we bought the first car that would be further developed over time. At the first milestone we are left alone with a car that can not be enhanced” [34].

The response on the part of Tesla was to offer the owners the option of trading-in their non-updated vehicles for a new one, at a loss, in the manner akin to a typical vehicle trade (i.e., to purchase a new vehicle). It should also be noted that not all Tesla owners agreed with the petition: a counter-petition aimed to “keep Tesla focused and on track” [35] (i.e., it encouraged the company to “ignore the demand from this group of owners who either aren't thinking clearly, or are being selfish”). Said one owner: “The more ‘revolt’ over this problem, the more I am inclined to agree that the people doing the revolting are acting as if they are, to put it mildly, feeling entitled” [36]. Said another owner: “Disregarding the ‘other’ petition will provide a clear signal that this kind of behavior is not what can and should be expected from early supporters” [37].

Looking at update unrest among Tesla owners evidences several things about update unrest activities more generally and about the particular role of DSM in the episodic power struggles that emerge between end users and software makers. First, the presence of update unrest among Tesla owners shows that update unrest activities crisscross socioeconomic groups and categories. Tesla owners are presently said to comprise a “very high-end niche market” due to the relatively small number of Tesla vehicles currently sold in a given year (35,000 in the U.S. in 2014) and due to the high cost of each vehicle, which not uncommonly reaches and exceeds $100,000USD [38]. Looking at update unrest among Tesla owners involves, then, looking at elite culture—at wealthy social actors with atypical levels of disposable income, suggesting that update unrest happens across social class lines.

Second, looking at update unrest among Tesla owners reveals the way software user expectations are being formulated across multiple, accreted experiences that involve having used over time a disparate range of hardware and software technologies, and that involve having engaged with an array of software updating practices. Participants within the Change.org petition repeatedly drew comparisons to practices surrounding other scales of mobile devices (e.g., smartphones, etc.) as a way to make sense of, and critique, Tesla’s actions, as well as drew comparisons to other update practices. As one owner put it: “I feel cheated and left out...This not a phone for crying out laud [sic]” [39]. Said another owner: “[I] can't throw a way the car like an Iphone!” [40]. Another owner put it this way: “Just like Apple's anticipation of products releases, which is announced months before schedule, Tesla should do the same as well. This is not a $700 device we're talking about, this is a $100,000 device” [41]. Noted another unrester: “At least many other ‘technology’ companies have the decency to announce upcoming technology advances so that a consumer has the option to wait to purchase” [42]. Such statements indicate that prior patterns of use and experience with other hardware and software update practices were, for some, a key factor behind their Tesla-focused unrest.

Third, the recent update unrest among Tesla owners shows that new DSM platforms can quickly enter the mix when it comes to facilitating update unrest activities. Over the past several years, Tesla
owners have popularized two distinct online venues, Tesla Forums and Tesla Motors Club, for communicating and exchanging information as a committed and quickly growing, but never stable, social unit. However, in the fall of 2014, in the wake of the unexpected updates, Tesla owners rapidly turned to a third, unrelated platform (Change.org) to engage in update unrest. What moving to Change.org seems to have specifically afforded Tesla owners is a place to communicate and exchange information not governed by the company itself, which is how Tesla Forums operates, and not tightly moderated by Tesla enthusiasts, as is the case with the Tesla Motors Club. Change.org also provided a more public venue for staging update unrest, given that existing DSM venues around Tesla ownership are not widely followed among the general public. Change.org also offered the added benefit of leaving the existing DSM platforms privileged by Tesla owners fairly undisturbed (i.e., maintaining community even during a time of unrest).

4. Discussion

We are witnessing the rise of a culture of persistent and pervasive updates. Readers familiar with personal computing software, Internet browsers, biotechnologies, automobiles, or wearing pants will recognize updates from everyday life. They can be as banal as an oil change or a font replacement, or stark like the oncoming transition from one cut of jeans to another. In line with persistent update culture, and perhaps mirroring it, entire categories of technology are now made by updates, including and foremost software. Most software technology users accept updates, internalize them, submit their data for analytics, and thereby further what have now become fairly established technology production processes in the software sector. But accepting and submitting to a new update almost inevitably leads to more updates. For this reason and more, software users also sometimes resist updates, which can include reduced-use, non-use, departure (i.e., software switching) [43], even rooting or jailbreaking devices at the risk of rendering the hardware useless altogether. It can also include active protest, critique and organizing—update unrest. This update unrest can likewise emerge when users perceive themselves to be left out of key updates or future updates. Update unrest is almost certainly a carry-over from older, pre-digital social and political practices (e.g., boycotts, mass petitions, group lobbying, tool breaking, walk-outs), yet also an emergent way of being imbricated with networked technologies, including DSM. Not only are updates and update unrest becoming part of daily life on DSM but, as the two cases presented above show, this update unrest culture is giving rise to new types of DSM user behavior and communities across DSM platforms. The 2015 updates to Tinder deepened ties to other DSM platforms but, according to many observers and users, this change erased the simplicity that made Tinder popular in the first place and undercut the basic functionality that led to its rapid adoption. In our case studies we found that update unrest exists as cross-platform phenomena. Users pushed back against the revamp update on other DSM platforms: providing thousands of reviews that reported initial release bugs, critiquing the adjoining platform integration, and lamenting the match limiting feature, which would thereafter police particular kinds of opportunistic use. Tinder users also publicly registered service slowdowns and critiqued the idea of trying to address a widely reported problem (e.g., spambots) by regulating user behavior in new ways. The late 2014 updates to Tesla involved a group of wealthy mobile device (i.e., “computer on wheels”) users who claim to have made the promise of future updates a key part of their purchasing decisions but then found themselves feeling left out when Tesla made a rapid series of rebuild updates that, to the surprise of many, would not be universally available. In response, Tesla users turned to DSM to stage a formal protest, channeling but also aggregating and amplifying their voice and frustration.

As with the case of the iOS6 Maps rebuild update, the Tinder and Tesla cases capture how, through DSM, people seem to be using update unrest to pool knowledge about software technology, to formulate ideas about how software technology is made, and to debate how their participation is enfolded into not just software development processes but also technology adoption and social media creation cycles. In the case of the Tinder update, DSM allowed people a venue for expressing conscientious objections to platform convergence and behavior modification, along with complaints about service slowdowns. Users were critical of new data collection features, and of what looked to be an effort to transform Tinder into more densely nested and interlocking media. In other instances, such as in the case of Tesla, update unrest suggests that a set of norms and expectations about update practices are emerging across user experiences; the case of Tesla is a case where accumulated user expectations were perceived to be suddenly violated.

Both cases underscore that update unrest varies. But the cases also suggest that update unrest may follow mappable patterns. In each case: initial service...
outages, delays, or downtime (common to new rollouts) were followed by initial user engagements with the software changes were followed by update unrest activities on adjacent DSM platforms that quickly intensified, like a social media spark or starburst in a slightly different social domain or sociotechnical galaxy. Extending this observation, we found that neighboring DSM platforms play an interesting role in the forms of unrest that we identified. The common role of DSM in these instances was to enable people to organize around specific updates. But the affordances, instabilities, and technology-in-the-making status of different DSM platforms helped shaped the contours of the ensuing update unrest activities. “Similar tools do not necessarily yield similar practices,” Ingrid Erickson reminds us [44]. In our study, we found one set of update unresters turning to Change.org, which is structured around a concrete statement that people sign and support, along with group discussion and commentary. We found another set of unresters, in the case of Tinder, privileging app store review forums that allow people to publicly evaluate technology, give comments, and rank products. And these were just the dominant outlets. On Twitter for example, when Tinder announced the revamp, users were able to directly address the announcement with @mentions to the company.

We also found that the underlying power dynamics between software developers and end users were ultimately sustained. Unlike the famous case of iOS6’s Maps, neither Tinder nor Tesla has yet to back down from the changes that we tracked and studied. In both cases, DSM was employed to facilitate update unrest but, in the end, the makers prevailed and the march of Tinder and Tesla software permutation continues onward.

5. Conclusion

This paper investigated the cultural phenomenon of software update unrest, focusing on two exemplars from the past 365 days that likewise highlight the growing centrality of DSM for negotiating power dynamics between software developers and end users. It would be too soon to say whether update unrest is evidence of a new kind of consciousness about software or signals shifting public understandings, morals and beliefs about network architecture and software development practices. Claims about individual and mass consciousness are tricky to make, and even trickier to defend. Moreover, patterns of software use are folded into software development in what, for the uninitiated, can seem like perplexing ways. User data is gathered by a range of intermediaries, including network service providers, device manufacturers, app developers, and mobile operating system administrators, amongst others. Typical terms of service for downloaded and factory set applications involve sharing your application usage, location data, and power usage. Usage patterns, data analytics, and recorded information behavior are commonly shared or co-owned. These patterns of practice create complex, heterogeneous, and structured trace data that are valuable to a range of actors involved in the further development of software technology. Whether update unrest signals growing awareness or consciousness about some or all of these aspects of the increasingly networked software technology production process would require further study to determine with confidence.

Questions for future research might include: Why does update unrest not happen in cases that seem likely to lead to unrest based on past update unrest occurrences? That digital social media seems to facilitate and enhance software update unrest is clear. Might DSM also limit it? Is it possible that something about DSM, despite its diversity, lends itself to sustaining power dynamics between software developers and end users in a manner that privileges the former, as seems to be the case in the recent unrest surrounding Tinder and Tesla? On a different track, does update unrest cluster around any particular software features or update approaches? Are there instances of update unrest that break from the starburst formation that seems to be present in both of the cases examined here? And what, if anything, might update unresters have in common with the types of information exchange and organizing, as well as circuits of co-dependence, that have emerged around digital laborers, like the Mechanical Turk workers studied by Lilly Irani [45] and others.

With update unrest in response to specific updates gaining increased prominence, including unrest about specific DSM platforms, we may be seeing the relationship between users and software technology in the early stages of fundamental change, and we may also be seeing the emergence of new models for dealing with update culture more broadly. While the well-known case of iOS6 Maps is important to keep in mind, this work identified a key gap in the existing literature and added new exemplars into it. This work also sought to show that beneath update unrest lies something potentially far more exciting and revealing than generalized conservatism among software users. At stake in studying update unrest, in our view, is not just matters of aesthetics and habits of use, rollout
blunders and customer relations, but also, and more importantly, remotely enacted, highly social, and heavily mediated debates over how technology is made.

6. References


[8] T. account, “GOOD NEWS: Everyone seems to be checking out their revamped Tinder profile. BAD NEWS: Tinder might be slow / down in the next 24h,” @Tinder, 15-Apr-2015.


[16] Amber, “Read tinder AppStore reviews for a good time pic.twitter.com/WRzRY4aASG,” @missamibear, 16-Apr-2015.


