Editor: **Jane Cleland-Huang**
DePaul University
jhuang@cs.depaul.edu

Requirements That Reflect Social Responsibility

Jane Cleland-Huang

RECENTLY WE'VE SEEN social apps and products such as Facebook, Snapchat, Twitter, Instagram, and Pinterest emerge and trend almost overnight. Many of these products measurably enhance and support societal values—for example, by fostering communication, enabling social networks, delivering personal services, or providing commerce platforms. Apps and social-networking sites have become so ubiquitous that we rely on them for many facets of our daily lives—to plan driving routes, reserve flights, view photos, track our exercise routines, or select and stream movies. Facebook founder Mark Zuckerberg recently claimed that one billion users, or one-seventh of the world's population, had used Facebook in a single day![1]

> Apps designed to benefit one particular user group might offend or harm others.

Potential for Harm

Unfortunately, the story isn't universally rosy. The same products that enhance our lifestyles also have the potential for significant harm.

In 2013, a 19-year-old man committed suicide after two young women wrote "malicious and completely unfounded" posts about him on Facebook.[2] Sadly, this incident isn't isolated, with several recent studies showing cyberbullying to be on the rise. The prevalence of other Internet-based crimes, such as cyberstalking and cyberharassment, has led many countries to enact specific laws or change existing harassment and stalking laws to address these antisocial behaviors.

More recently, the Peeple app was announced. Instead of collecting ratings of businesses, hotels, restaurants, or movies, Peeple would collect ratings on individual people. Users would grade the personality of their friends, ex-boyfriends, teachers, bosses—just about anyone they came into contact with. Peeple's cofounder, Julia Cordray, explained that once someone entered a review into the system, the subject of that review wouldn't be able to have it removed. The premise was that both positive and negative reviews were important for the app's success and utility.

However, the idea of distilling someone's personality into a five-star rating led to public outcry. This in turn resulted in immediate backtracking, which I discuss in more detail later. It's uncertain what form the app will assume when it eventually, if ever, hits the market.

In another example, SketchFactor was released in 2014 as a "community empowerment" tool that would let people record firsthand reports of sketchy incidents. For example, users might report a homeless person sleeping under a bridge, a location at which they felt harassed by a construction crew, or a corner at which they noticed an "insalubrious" group of young people congregating. Critics quickly raised issues of racial profiling and pointed out that sketchiness is very much in the eye of the beholder. Furthermore, profiling a neighborhood in such a way would surely be offensive to its residents and would further disenfranchise and adversely affect a neighborhood that's probably already down on its luck. Whether you agree with

0740-7459/16/$33.00 © 2016 IEEE

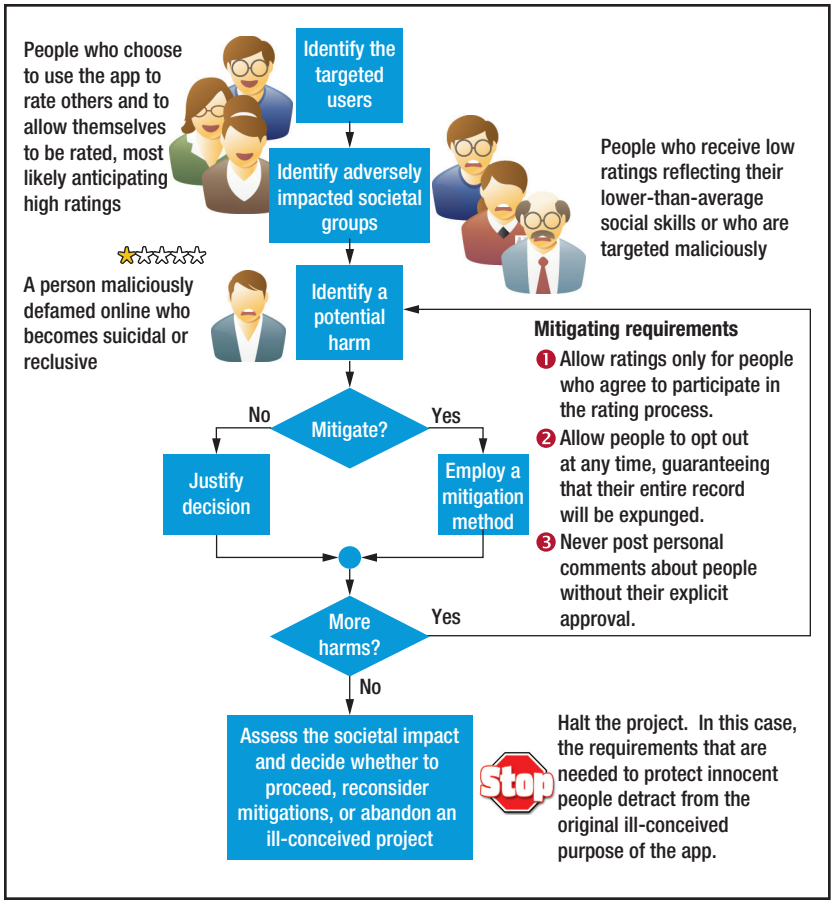

FIGURE 1. A socially responsible requirements-engineering process for creating an app for rating people.

SketchFactor's cofounders or critics, the fact remains that apps designed to benefit one particular user group might offend or harm others.

Who's Responsible?

The critical question is whether designers and developers should bear responsibility for the social impact of the products they create. In an interview with *Bloomberg Business*, Jim Adler said that "we [developers] can make stuff work—it's not our job to figure out if it's right or not."[3] I beg to differ. Product developers can, and must, consider the social and ethical ramifications of products they deliver.

But how do we do that? To answer this question, let's take a closer look at how requirements are discovered and dealt with in the fast-moving, fluid world of app development.

Injecting Ethics into Design Thinking

In many app projects, the traditional requirements-engineering process takes on the form of feature design and selection, with app founders, developers, funders, and users assuming the responsibility for feature and process vetting. However, if we accept that many new products on the market are developed without formal requirements elicitation and analysis, it's important to define new commonsense practices for ensuring that ethical requirements are considered from the get-go. These are explicitly defined requirements that protect the needs of vulnerable and potentially adversely impacted stakeholders.

Figure 1 shows such a socially responsible requirements-engineering process for a rating app like Peeple. After identifying the targeted users, we identify stakeholder groups that the product might adversely impact. These often aren't difficult to imagine—for example, people whose reputations are maligned through low personality ratings. Once we've identified those stakeholders, we can systematically identify the ways they could be harmed—for example, by privacy intrusions, loss of personal reputation, or product addiction. (An example of such addiction is when a young Korean couple became so engrossed in raising their online virtual child that their real three-month-old infant starved to death.[4]) Only when we've documented the potential harms can we identify mitigations and ultimately decide whether the app is socially responsible and financially viable.

Identifying social impacts can be difficult. Securing a software system is sometimes described as a game of cat and mouse in which developers adopt current best security practices, only for the system to be attacked later by a new, previously unknown, attack mechanism. Building socially responsible apps suffers from the same challenge. New products change how people behave; sometimes the new behavior leads to previously unimagined social impacts, which must be addressed as they emerge.

Although some people might consider it naive to expect product innovators to take the time to analyze social impact, recent history suggests that peer pressure will eventually drive them to do so anyway. In the case of Peeple, the cofounders reacted to the public outcry by creating requirements on the fly. First, they declared that all ratings would be posted, positive or negative. Then, they declared a set of requirements that allowed people being rated to review and dispute negative posts and disallowed negative posts against people who had never logged into the system. Finally, in a last-ditch effort to save their product, they announced their expectation that people's good nature would shine through and that Peeple would turn out to be overtly positive. Much time, effort, and ill-feeling could have been averted if they had thought through the app's social ramifications in advance.

Fortunately, most app developers desire to deliver socially responsible apps. The few simple steps I outlined here could help us think through potential harm, identify mitigating requirements, and ensure that the products we develop maximize good and minimize harm.

References

1. S. Dredge, "Zuckerberg: One in Seven People on the Planet Used Facebook on Monday," *Guardian*, 28 Aug. 2015; www.theguardian.com/technology/2015/aug/27/facebook-1bn-users-day-mark-zuckerberg.
2. H. Arkell, "Coroner Warns of Dangers of Facebook after Student, 19, Targeted by Young Women Bullies Online Hanged Himself," *Daily Mail*, 26 Nov. 2013; www.dailymail.co.uk/news/article-2513782/Facebook-bullies-led-suicide-student-19-hanged-himself.html.
3. J. Robertson, "How Big Data Could Help Identify the Next Felon—or Blame the Wrong Guy" (interview with Jim Adler), *Bloomberg Business*, 2013; www.bloomberg.com/news/2013-08-14/how-big-data-could-help-identify-the-next-felon-or-blame-the-wrong-guy.html.
4. M. Tran, "Girl Starved to Death While Parents Raised Virtual Child in Online Game," *Guardian*, 5 Mar. 2010; www.theguardian.com/world/2010/mar/05/korean-girl-starved-online-game.

JANE CLELAND-HUANG is a professor of software engineering at DePaul University. Contact her at jhuang@cs.depaul.edu.

See www.computer.org/software-multimedia for multimedia content related to this article.