AS RESEARCHERS AND teachers and practitioners, we “software types” excel at multitasking. This, in part, led us to ask the question: Can one attend a software engineering conference and do something good for society? We found the answer to be a resounding yes. In this article, we present our first experience of running RE Cares, a conference collocated event. This event included a workshop, conference sessions, and a hackathon for developing an application to support emergency field activity for Mutual Aid Alberta, a nonprofit organization coordinating natural disaster responses in the Canadian province.

Toward Software for Social Good

Computer scientists and software engineers can give back to the community and participate in “public good” projects in many ways, from donating to the causes they find worthy, to volunteering their time, to championing causes. Yet, unlike the proverbial “doctors and lawyers” who can contribute to public good in straightforward ways that specifically take advantage of their professional expertise, such opportunities for computing researchers and professionals are not immediate. Our group set out to address this.

At the IEEE International Requirements Engineering Conference 2017 (RE 2017) held in Lisbon, the presentation by Guenther Ruhe, Maleknaz Nayebi, and Christof Ebert about the role of RE in society sparked an active discussion among the conference participants about their ability to “give back” in a professional capacity. At the end, a group of us said “we want to try,” and RE Cares was born.

The Concept Is Simple

To achieve its goal, RE Cares needed to

• **Step 1**: target a conference.
• **Step 2**: find a local stakeholder with need of a software product that falls under the broad definitions of “public good.”
• **Step 3**: organize a requirements engineering, design, and prototype-building hackathon for the stakeholder’s application during the conference.
• **Step 4**: pass the results of the hackathon to the open source community, and organize a team to complete development.
• **Step 5**: deliver to the stakeholder. As simple as the idea sounds, it is fraught with uncertainty. Can we get the conference to collaborate? Can we find the right stakeholders? Will conference attendees participate, engage, and buy into such an idea? Will we have the critical mass of participants to achieve something useful? How does one go about running a requirements engineering hackathon? How can we keep the torch going after the conference is
over? All of these are important and valid questions, and all of these were considered. And yet, we still decided that we wanted to try.

As RE 2017 gave birth to this idea, we chose RE 2018, scheduled for August 2018 in Banff, as our place to start. The conference name helped give us the name for the event: “RE Cares.” With nothing but an idea and a name, we started our undertaking.

Preparation
Early in developing RE Cares, the idea of helping a local stakeholder took shape—in a sense, we wanted the largest gathering of requirements engineers in the world to give back to the host region. We wondered, what can we do in Alberta, a province in Canada known for agriculture, oil industry, and two rival hockey teams?

One of our organizers put us in touch with Mutual Aid Alberta, a nonprofit organization that supports Alberta’s disaster response organizations with response tools and training. Our stakeholders, Shell Clarke, the chairman of Mutual Aid Alberta, and Chuck Brophy, a systems engineer working with Mutual Aid Alberta on its web portal, have produced a long list of software ideas and software needs that have not been met. With their enthusiastic support, we commenced our preparations for RE Cares.

Starting in early October 2017, the RE Cares idea was pitched to the organizing committee of RE 2018. By January 2018, we had secured the aforementioned stakeholders in Alberta and had secured a half-day International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ) Cares workshop slot at REFSQ 2018 to “dry run” RE Cares.

Monthly telecons with the stakeholders and organizers began in March, transitioned to biweekly calls in June, and then to weekly calls in the few weeks preceding RE 2018. In these telecons, we were able to hash out the general direction of the requirements engineering and software development effort. To support efficient disaster response, Mutual Aid Alberta proposed the development of a web-based instant messaging system that would allow incident response commanders to communicate, both formally and informally, with all first responders engaged in the incident response through an established chain of command. We established that existing off-the-shelf messaging systems lacked proper functionality and reliability in the conditions of disaster response, making a specialized software solution a reasonable choice.

In Banff
RE 2018 took place 20–24 August 2018 in Banff, an idyllic ski resort in the Canadian Rockies, nestled inside a national park and surrounded by mountain peaks and lakes. Yet the British Columbia forest fires served as the most visible conference backdrop, shrouding the surrounding area, famous for its crystal clear air, with a thick layer of smog and raising the stakes for the success of RE Cares—committed to helping disaster response personnel deal with the very events we were observing.

RE Cares was introduced into the RE 2018 program as a Tuesday full-day workshop followed by an official event Wednesday afternoon (in the RE 2018 program) and a series of unofficial events on Wednesday and Thursday culminating with a short report back to the conference on Friday during the closing remarks.

On Tuesday, a group of about 30 conference attendees, consisting of the RE Cares organizers, RE academics and practitioners, students, and stakeholders, met to discuss the software (see Figure 1). In the morning meet-and-greet session, we introduced the event, discussed
the logistics, and had two key presentations: the organizers gave the stakeholders a short overview of requirements engineering as a discipline and of the requirements elicitation process, while the stakeholders introduced the event participants to Mutual Aid Alberta, its role in disaster preparedness training and incident responses, and its software needs. In the follow-up Q&A session with the stakeholders, an informal process of requirements elicitation commenced. The majority of the questions in the session, captured together with their answers in the ever-growing requirements elicitation document, revolved around the restrictions on the software functionality (what is included and what is not), the discussions of the roles that the software users would take, and the challenges that the software users (the first responders working on the sites of the incidents, often without Internet and cell phone access) may experience when working with a web-based application.

Tuesday afternoon sessions involved a creativity exercise and demonstrations of two techniques to elicit requirements: task-driven approach and design-thinking approach, led by Barbara Paech and Meira Levy, respectively.

The Wednesday afternoon panel was the first truly public introduction of RE Cares to the conference. With new people in the audience, the panel involved a short presentation by the stakeholders about their software needs and a newly invigorated Q&A session, during which new issues were raised (and new requirements documented), together with continuations of discussions that started on Tuesday. Following the panel, a small group of RE Cares participants formed two teams: one team worked with Shell Clarke to design the use cases and the user interface for the instant messenger application, while the second team worked with Chuck Brophy to develop the relational database model for the application.

Thursday was reserved for an all-day hackathon during which a number of conference attendees, spearheaded by University of Kentucky developers, worked on building software prototypes (see Figure 2).

In parallel with RE Cares, a satellite event took place in Israel at the Shenkar College of Engineering and Design. There, three students undertook a four-day hackathon and applied design thinking to develop Ranger, an Internet of Things device for detecting forest fires, alerting authorities, and navigating a hiker out of the forest away from the fire. The students developed a design prototype of the device as well as a video explaining the concept and design. The video can be found on the RE Cares website (wrecares.wixsite.com/recares).

Lessons Learned

No worthwhile undertaking is without its shortcomings (and successes), and RE Cares was no exception. Our lessons learned come from two sources: 1) a survey of organizers, stakeholders, and participants and 2) the organizers’ own feedback to augment the survey findings.

Our first lesson is that running RE Cares is a serious logistical challenge. It requires early buy-in from the host conference, proper understanding of the exact structure of the event ahead of time, and proper allocation of resources to the event throughout the duration of the conference.

Before the conference, as we did not quite understand all the moving pieces of our own concept yet, we could not elucidate our room needs early enough to the very accommodating organizers of RE 2018. As a result, in the end we had to structure RE Cares to reflect the room availability.

We did not get the level of on-site participation of developers at RE 2018 that we had hoped—we learned that we must bring more developers with us or have firm commitments of the developers (and possibly their advisor) beforehand.

FIGURE 2. The working sessions.
ABOUT THE AUTHORS

ALEX DEKHTYAR is a professor at California Polytechnic University, San Luis Obispo. Contact him at dekhtyar@calpoly.edu.

JENNIFER HORKOFF is an assistant professor at Chalmers and the University of Gothenburg. Contact her at Jennifer.horkoff@gu.se.

JANE HUFFMAN HAYES is a professor and associate chair of the Computer Science Department at the University of Kentucky. Contact her at hayes@cs.uky.edu.

MEIRA LEVY is a senior lecturer at Shenkar College of Engineering and Design. Contact her at lmeira@shenkar.ac.il.

IRIT HADAR is a senior lecturer at the University of Haifa. Contact her at hadari@is.haifa.ac.il.

MALEKNAZ NAYEBI is an assistant professor at Ecole Polytechnique de Montreal. Contact her at mnayebi@polymtl.ca.

ERIN COMBS is a software engineer at Lexmark. Contact her at erin.combs@lexmark.com.

BARBARA PAECH is chair of Software Engineering at Heidelberg University. Contact her at paech@informatik.uni-heidelberg.de.

ALESSIO FERRARI is a research scientist at Consiglio Nazionale delle Ricerche, Istituto di Scienza e Tecnologie dell’Informazione “A. Faedo,” Italy. Contact him at alessio.ferrari@isti.cnr.it.

JARED PAYNE is a student at the University of Kentucky. Contact him at jmpa235@uky.edu.

SARAH GREGORY is a requirements and systems engineer at Intel Corporation. Contact her at sarah.c.gregory@ieee.org.

MATT PRIMROSE is a functional safety engineer at Intel Corporation. Contact him at matt.c.primrose@intel.com.
We did a nice job of getting what little funding we needed: the RE 2018 organizers allowed our stakeholders to attend for free and also allowed the students to attend the workshop days for free, the University of Kentucky Industrial Partner Association provided T-shirts for all participants, and Lexmark paid for one of the developers to travel to the conference.

During the conference, the overarching goal was to get people into the event, keep them there, and make them productive participants. While there was some interest in the event (20–30 people at all sessions except the hackathon), we would have liked to see more people attend and work with us. We spent too much time in the initial stages of elicitation, getting to breakout sessions and design only at the end of the conference day on Wednesday (ideally, that session should have happened significantly earlier in the day). We also learned that our hackathon process could be streamlined; we ended up trying to do too much in the time that we had.

The key after-the-conference lesson is that the RE Cares work does not end with the conference closing session. The design sessions and the hackathon left artifacts that have to be completed, and in some cases improved, before delivery to the stakeholders.

Our major lesson, though, is simple and straightforward: our efforts paid off! RE Cares received support, buy-in, and appreciation from stakeholders, conference organizers, and, most importantly, conference participants. Some additional benefits have

(continued on page 94)
immutable data structures and pure functions into all kinds of languages, often enforced by the compiler. I see a connection between Brooks’ drive for conceptual integrity and Rich Hickey’s advice on how to find simplicity in design.

Across the decades, formalism has helped us think about our designs, and it’s increasingly accessible. Where Dijkstra would have used a pencil and paper to formalize his thinking about a program, a programmer today following best practices might build abstractions bottom up with clear contracts and invariants, structure the overall system according to architectural patterns, and use the compiler and static analysis to ensure each of the user-defined types fit together as expected.

My intent with this column is to make it easier to talk about intellectual control and connect it with practices we already use on our projects. It’s difficult to talk about the kind of control we have over our software and more difficult still to talk about different approaches like intellectual control and statistical control. I’m thankful for Rich Hickey’s metaphor of driving a car without hitting the guardrails as it brings this topic to life.

Software is everywhere, and it’s a common lament that it’s big and buggy. As the software gets bigger and bigger, it’s more and more tempting to settle for statistical control with tests. Nobody wants to write buggy software, but many don’t know another way or how to avoid analysis paralysis. You can choose a few places in your code, build up your understanding to gain intellectual control, and share that understanding with your team. Soon you will be driving your car without fear of hitting the guardrails.

Reference