I would not call it a revolution. It is more about making the whole system work in balance. If we use a technical metaphor, nowadays in Russia some processes in the system are so heavy that others do not have enough resources to run normally. As computer scientists, what we try to do is to clear cache and restart, so to speak, but without destroying everything.

—ALEXEY P., computer scientist, developer

When, in 2010, wildfires started burning near Moscow, governmental agencies were unable to quickly respond to the emergency and the official media simply downplayed or even silenced news of the disaster. Instead, it was the online community that came up with an alternative and efficient handling of the crisis (Asmolov 2010). Developing a web application based on the well-known crisis-mapping platform Ushahidi, a group of programmers, bloggers, and activists helped thousands of victims of the wildfires by successfully coordinating volunteers from all across Russia and consolidating a self-organized community (Machleder and Asmolov 2011). With about 170,000 unique visitors, it became the first mass-used Russian civic application—one that launched a wave of similar projects aimed at responding to important political, social, and economic challenges with the help of volunteer coders and crowdsourcing technologies.

In 2012 alone, 272 civic applications were developed in Russia. While in the United States and Europe civic software projects are typically part of “open government” programs and are largely supported by the state (Eyler-Werve and Carlson 2012), Russian civic hacking initiatives started as experiments without governmental support; they are bottom-up collaborations of hundreds of volunteer developers, designers, computer scientists, and activists. It is therefore important
to understand why, in the particular social and political context of contemporary Russia, civic hackers believe in coding as a way to “fix the system” without governmental support. Many political and cultural commentators stress the existence of an apathy and distrust toward political participation in contemporary Russia (Erpyleva and Magun 2014; Kharkhordin 2011; Zhuravlev, Savelyeva, and Erpyleva 2014).

However, that does not apply to Russian civic hackers. Conscious of the failures of the Russian government and well aware of the problems of corruption, inefficient city services, and electoral fraud, they actively discuss politics online, propose their own solutions, and even put forward visions of political parties driven by “geeks.” They refuse to engage in traditional forms of protest like rallies, strikes, and petition campaigns (which they consider to be inefficient), trying instead to invent, experiment, and tinker with new tools—assemblages of programming code and law—that have the potential to fight political apathy and improve the everyday lives of Russian citizens.

It would be wrong, however, to see civic hackers as revolutionaries or radical activists. A phrase found on a programmers’ forum—“We do not need revolutionaries, we need legitimacy”—could function well as this movement’s motto. Civic hackers operate clearly within the legal arena, developing civic apps as techno-juridical instruments aimed at raising government transparency to compel municipal civil servants to respect the law. Paradoxically, however, the demand for transparency is itself a radical act within the Russian context. Giving citizens the ability to surveil and control the administration, civic apps provoke strong and controversial reactions from governmental institutions, ranging from hostility to collaboration.

The Russian civic hacking movement is a complex and changing network of people, machines, servers, cables, NGOs, and administrative institutions. It is highly decentralized, with several active centers (IT communities and specialized NGOs) but no lead institution. As the interviewee quoted in the epigraph puts it, the Russian civic hacking movement “is not a revolution”—an observation we need to keep in mind to avoid all the possible romantic, nihilist, and maximalist connotations of terms like “hacker” and “civic.”

First of all, in this context “hacking” is not about coding complex systems, inventing new languages, or committing cybercrimes. Here hacking stands for tinkering and experimenting, a meaning close to that of the French word *bricolage*; the Russian term *smekalka* captures it as well. Derived from an old verb, *smekat’* (“to understand” or, colloquially, “to get it”), smekalka is a quality that Russians claim for themselves. Part of the “national character,” it is often mentioned in fairy tales, where it functions almost as a preternatural
power to find a solution quickly, in very tight spots, when one has no proper tools or means and can only use what is at hand. Civic hackers are those who can quickly invent a new, intelligent, and cheap way to solve social problems by means of information and communications technology (ICT).

As I will discuss in a moment, civic applications are often based on the integration of existing platforms, the reuse of existing texts (in particular legal texts), and pieces of code. Useful and popular civic applications successfully translate a feeling of social injustice (a “trouble”) into a public problem, and then match it with a suitable techno-juridical solution.

The Russian civic hacking movement is not about “geeks only.” Unlike computer science, which is rarely open to non-tech-savvy actors, the civic hacking movement is a hybrid field where subject-matter experts (NGO activists, legal experts, journalists, urbanists, or what Harry Collins [2002] calls “experience based experts”) can contribute to the production of software. What Collins called the “core-set” is thus relatively open to new kinds of publics, previously kept outside computer science and coding activities. All the Russian civic applications that I have studied are the result of collaborations of tech and social experts. They involve amateurs, beta testers, reviewers, and active users who contribute to the improvement of the software with their user experience, feedback, evaluation, and volunteer coding. While focused on very specific problems, these new networks have far-reaching effects: they restructure Russian civil society, help to overcome the isolation of tech-savvy people, and raise the computer literacy of social activists.

This chapter describes the civic hacking scene and its practices, but also its tensions and limits. Contrary to the belief in a general “apptivization” of societies, there cannot be an app for everything: Russian computer scientists and developers engaged in coding civic software often fail to produce significant social changes with their tools. Still, even if not all civic applications are ultimately successful, the civic hacking community is having an important impact on both Russian society and Russian computer science. Through the proliferation of different civic software projects, Russia can be seen as a political laboratory, where civic hackers experiment with proposing new technical solutions to social challenges and collaborate with new formats of political movements. Thus, two of the most used civic applications have been launched and financed by Putin’s chief antagonist, Alexey Navalny. In the last five years, this movement has managed to build an active and prolific community of activists, developers, designers, geeks, and lawyers who contribute to create value, share skills, democratize code, and engage coders in social issues.
While the diaspora of Russian computer scientists seems to actively reclaim its “Russianness,” Russian civic hackers refer to the transnational culture of free and open source software (foss) developed beyond national borders and identities. Programming languages are believed to be a universal grammar that can help to solve any kind of problem: “The whole world speaking one language—that is the power of digital code” (Joyce 2010). Many of the Russian civic hackers I have met connected their inspirations to the foss culture. Vitaliy V., the organizer of the annual hackathon “Spb Data Hack” and a computer scientist from ITMO University, attributes the very idea of using code to solve social problems to foss culture, quoting Don Tapscott and Anthony D. Williams’s Wikinomics (2006) as the main source of inspiration for his own work as a civic hacker: “It is about crowdsourcing and how a large number of people can help you to solve your challenge when you share your knowledge with others. Since I’ve read it, I started being interested in open innovation and in civic activism.” Most of the civic application projects relied on decentralized teams without physical offices or stable workplaces—in the interview, Vyacheslav K., developer of the RosYama app, called them “virtual teams.” All of my interviews took place in cafés or coworking spaces, except for those working with the team of Alexey Navalny. Coders emphasize this extreme mobility and independence from a precise physical location as a positive trend, which they oppose to the conservatism and stagnation of Russian society. A particular “imaginary” is thus being produced where Russian administration and its slow and complicated procedures are contrasted with a “new” global society without borders, where people, objects, skills, and ideas can circulate freely, just as software does within the foss culture. Within this imaginary, their modern, user-friendly applications contrast with the endless dusty paperwork of the Russian bureaucracy:

I feel pity for my parents who are stuck in these conditions [they live in the town of Bryansk]. There’s nothing there. When I go to see them in Bryansk, they say there’s nothing new going on. And I feel a bit ashamed to talk about my trips to the USA, Thailand, or Bali. . . . I think life is hell in the [Russian] provinces. And my goal is to show, on my own example, how people live in the contemporary world, one can work without being attached to a precise place, it is an incredible freedom, you can be everywhere and work from anywhere. (Tatyana, UX designer, WebNabludatel app)
My interviewees are inspired by foreign civic hacking examples, quote details and figures of foreign social innovation projects, and are curious about the latest examples of successfully functional civic apps. They claim to have good contacts with colleagues working in Kenya (with the Ushahidi project) and to participate in different RR events in India, Europe, and Canada. Some even collaborated with the hackers from the Indignados movement in Spain.

Code circulates between projects, and—in accordance with the culture of sharing of the FOSS movement—there are direct and indirect borrowings of pieces of code between Russia and the world. For instance, the founders of RosYama claim to have been inspired by the Fix My Street application that was launched in 2007 in Great Britain, giving citizens the ability to report urban problems to their municipalities. As an open source product, this application was installed in seven countries and credited with the solution of more than 250,000 problems. RosYama, an open source project itself, was developed in Russia, but as a result of being made openly accessible on GitHub, its code was rapidly borrowed by activists in the Ukraine, Belorussia, and Kazakhstan. It is said to have helped fix more than twenty thousand potholes across the former Soviet Union. Krasivyy Peterburg (KP), the application that helps report on different kinds of urban problems, has also been reused in more than twenty Russian cities. One of the developers on the KP team explains the importance of open source to civic innovation:

To keep the code open means that your product can always be improved, anytime, by anyone. You win from it, because some guys just come and say—I see you have a bug here, I can fix it, just for fun. And they can also borrow your code and bring it to their country and do some good. The social good can’t have an owner, it would be absurd not to share your code with someone who wants to improve people’s life.

Special events are regularly held to bring coders, activists, and politicians from different countries to work together on major global challenges. Since 2010, events like hackathons (forty-eight-hour coding competitions) or bar-camps (informal conferences of IT professionals and entrepreneurs) have become more and more popular in Russia. In September 2011, a binational hackathon “Code for Country” was held simultaneously in Moscow and Washington, DC, co-organized by Russian and American partners. On the American side, it was Emily Parker from the US State Department who organized the competition, presenting it as part of a project to “come up with creative ideas for digital diplomacy, looking for opportunities for Russia-US
cooperation” (Emily Parker, interview conducted by the author). The theme of the hackathon was “transparency and corruption,” expecting teams from both countries to code applications that could creatively respond to those challenges. In contexts like these, programming code becomes a tool of diplomacy and peace building, an instrument that can effectively work beyond the dominant geopolitical power relations.

It was at “Code for Country” that Alexey P., head of the IT company Progress Engine, worked on a civic project for the first time, providing “the first bifurcation point that converted [Alexey] into doing civic apps.” Months later, he developed the WebNabladatel mobile application for the monitoring of elections (Ermoshina 2014b). While the applications prototyped during this event were not released on app stores, they had an important impact on the Russian civic hacking community by helping to establish connections between civic activists and developers.

Since 2012, dozens of other civic hackathons have been held all around Russia, both in Moscow and Saint Petersburg as well as in the provinces, in an effort to decentralize and democratize innovation. They include “Hack against Poverty,” “Hack against Corruption,” “Social Impact Hack,” “Crowdlab,” “Apps 4 All,” and others. Unlike high-tech industry in the regions of Kazan or Siberia (Alina Kontareva, Andrey Indukaev, this volume), where decentralization happens via a state-driven process, through an important governmental intervention, in the case of civic apps an effort to bring social innovation to the remote regions of Russia has been pushed forward by NGOs. For instance, Teplitsa Sotsial’nykh Technologiy (the Greenhouse of Social Technologies) organized civic hackathons in the cities of Vladivostok, Krasnoyarsk, Yekaterinburg, Kazan, Barnaul, Samara, Ulyanovsk, Novosibirsk, Penza, and so on. These civic hackathons build an all-Russian network of IT volunteers, connect civic activists, tech entrepreneurs, and coders, and establish direct links between programming code and problem solving. This results in a peculiar sociotechnical network that operates in parallel to the state using code as a means of coordination. The organizers of these events believe that code can and should be used to address problems that have been previously delegated to human agents, social movements, or traditional political institutions. As Lilly Irani has argued, the civic hacking movement is not only about producing software prototypes but also about developing a specific “ethos”: “They manufacture urgency and an optimism that bursts of doing and making can change the world . . . and imagine themselves as agents of social progress through software” (Irani 2015). But
is this “optimism” enough of a motivation to work for free? Indeed, a question quickly arises: who pays for civic apps? What is the economic context of their production?

CODING UTOPIAS: CIVIC HACKERS IN SEARCH OF AN “OTHER RUSSIA”

According to Nicolas Auray, the free and open source software movement is marked by a “massive and brutal rejection of commercialization of social relations. For these hackers the basic organizational factor in their lives is not money or work but their passion for code and the desire to create with others something that has a social value” (Auray 2002). Vyacheslav K., developer of the widely used mobile application RosYama, agrees with Auray:

We always wanted to create something for ourselves, even if this may seem naive, we wanted to be useful to society . . . because our commercial projects are very cool, we can eat thanks to these projects, but sometimes one needs to build something “for the soul,” we wanted to bring our little contribution. And we worked on this project for free. It is, so to say, a gift to society.

All civic applications I observed were developed as side projects: developers worked on them apart from their official jobs as programmers. This particular mode of existence refers back to the specificities of Russian technological entrepreneurship mentioned in the introduction to this volume, with its characteristic “gap between the level of available mathematical, technical, and coding skills and the familiarity with entrepreneurial practices and culture.” Only when projects become very successful, and an environment is set up with a proper entrepreneurial infrastructure, can a developer be paid to work full time on a civic application. This requires the intervention of a new kind of third-sector organization, which acts as an “incubator” for civic technologies. Such is the case with the Foundation against Corruption (Fond Bor’by s Korruptsiyey), the ngo founded by Alexey Navalny, today one of the main opposition leaders, which has several developers on staff for the maintenance and updates of their three civic apps: RosYama, RosPil, and RosZKH. These developers, however, are paid not for coding a product from scratch, but for technical maintenance, bug fixing, and updates.

It is interesting that the professionalization of civic hacking tends to occur when the innovation has become more or less stable; when it has already
found loyal users it can rely on. However, in the early stage, civic projects emerge and are built on a volunteer basis, as “stigmergic collaborations” (Eliott 2006)—a concept used to describe a spontaneous self-organization within open source and civic-hacking communities based on indirect incentives and mutual support (Eliott 2006; Gregorio 2002; Heylighen 2007b; Parunak 2006; Robles, Merelo, and Gonzalez-Barahona 2005). The projects I have studied were launched as decentralized collaborations of individuals through GitHub, HabraHabr, Stake Overflow, Livejournal, mailing lists, groups in Vkontakte, and other online ecosystems, without a predetermined technical task or a strict division of labor. Typically, an idea is proposed on a blog, a team is quickly formed, a page on GitHub is opened, and the work starts with crucial input from the broader IT community. Users and testers become coproducers and participate in the maintenance and improvement of the project.

The collaborative and stigmergic nature of civic software extends to its business model. The apps I have studied were either supported by crowdfunding (RosYama, RosZKH, RosPil), grants and prizes (Krasivyy Petersburg, which won the Kudrin’s Innovation Prize), or financial support from IT entrepreneurs. Globally, no efficient and sustainable business model has yet been found for civic applications. This supports the thesis on “lack of conditions that facilitate technological innovation” outlined by Marina Fedorova in her chapter on Yandex (this volume). Instead, the civic hacking scene operates on “enthusiasm.” My respondents claim to work “for themselves” and “for the soul,” but also “for the country” and “for the people.” For instance, Timofey T. (the backend developer of WebNabludatel) explains:

I thought it was the best thing I could do to fight against . . . how to say . . . the regime. Really, without joking. I could have said that it was just for fair elections, but these are just words. It was the only thing I could do to fight against the regime. And I worked for free, I even gave a hundred bucks to pay for the servers.

For others, the motivation comes from a mix of political commitment and technical challenge. The “nonprofit gusto” described by Andrey Indukaev in his chapter on Tomsk (this volume), is a very relevant trait for civic hackers who share with Siberian computer scientists a specific attraction to problem solving, where technical complexity and social importance of a given task constitute a more important value than the monetization of the work. As Alexander M. (developer of OpenSalary, an application that provides information on teachers’ salaries) puts it:
I was always interested in treating big data. It is my passion. And I adore maps as well. So I had a big technical interest in this project. When they [the teachers’ trade union, “Utchitel”] proposed this project to me, I also wanted to help them because it is a shame that teachers have such miserable salaries and there is no open data on how much they must really earn. I felt pity for them, as they wanted to do something but had no idea of how the code works. And also . . . there are lies everywhere in our country. And I am very tired of this.

Often the coders have difficulty distinguishing between their passion for code, their interest in a technical challenge, and the urgency of a social issue. For them, coding becomes their way to participate in producing social and political change in Russia. This form of engagement through code is also related to a certain mistrust toward traditional forms of public participation. Olga, a former member of Gov2People and manager of several civic IT projects, explains this choice:

I went to some demonstrations but I do not really believe in this kind of actions. People go there to meet other people, to chat with them, to take selfies, this is me with Navalny or this is me with OMON [riot police], and so on. But it does not really change things, that’s my opinion. So I prefer coding some projects that can really help people. For example, in 2011 when there were all these big demos, we with Alexey P. made an app HelpWall that aggregates tweets and helps people who get arrested to get legal help or food and warm clothes.

While telling me the same story, Alexey P. added: “I do what I know best and if it can help someone in Russia, that’s great. Some people are good on the streets, some people are good on their laptops.”

Paradoxically, despite the transnational character of FOSS and the civic hacking movement, the Russian civic hacker community shares a certain type of patriotism: a loyalty to a particular idea of Russia—an idea strong enough to make them code for free for months and even contribute their own money to maintain the servers and update the applications. Civic hacking can thus be opposed to the “brain drain” trend in a peculiar way: even though civic hackers circulate a lot, they still code “for Russia” from abroad. Inventing new ways to use code to improve life in Russia, developers contribute to coding utopias and sustain their vision of a better Russia. What they share is a will to build a system where law would be respected, where existing public institutions would work properly and without corruption, and where citizens would have a means to control their administrations and
make them accountable. They code for the country or for the people, but not for the government:

Actually what I really want is that institutions work properly, that they execute the functions they have been made for. If it becomes true, if the police work to prosecute real criminals, if the road inspection works to build the roads without spending four times more money than what is needed, and so on. . . . If it works like this, I do not care who our president is because, well, the state is a structure built to serve the people and not vice versa. (Vyacheslav, developer of RosYama mobile app)

Even though civic hacking “is not a revolution,” it paradoxically overlaps with antiregime politics. Two of the most used civic apps in Russia (RosYama and RosZKH) are being pushed forward by a politician who is running for presidential elections at the time of writing this chapter. Alexey Navalny is the first political figure who has based a large part of his political capital on these new technologies. RosYama was the first project that made him popular among “lay users,” as it addresses a crucial Russian problem known to all citizens: the quality of the roads. Instead of projecting a “big narrative” or global political ideology that could unite people from above, these pieces of software address precise and sometimes seemingly small problems from everyday experience and try to fix them, thus grasping the very particular post-Soviet spirit of “society of repair and maintenance” (Gerasimova and Chuikina 2004). This makes civic apps a very relevant tool in the post-Soviet context with its “distrust” in traditional politics and political ideologies (the feature previously described in this chapter).

PUTTING PROBLEMS INTO CODE

Civic hackers tend to present code as a means to solve the problems of Russian society, but how exactly does the translation from a social cause to strings and functions of Python, Java, or C++ occur? Can we really make an app for everything? And how do apps transform the way citizens define their problems, delegate responsibilities, and communicate with governments?

I followed several civic hacking teams (described in more detail in table 3.1) as they built and tested apps to monitor elections, denounce corruption, send out alerts about leaking roofs, broken street lamps, potholes, and so on. But what do these projects share besides the technical format—the app—through which they are addressed? It was by tracking these proj-
ects back to their sources (prototypes, ideas, drafts, discussions) that I discovered some striking similarities—similarities of experience, not structure. All of them started from the experience of a “trouble”: electoral fraud, urban problems, leaking roofs, corruption, and so on.

Traditional responses did not, however, give satisfying results, as official institutions proved unable to adequately address the problems. As a result, the authors had to launch an inquiry (in the pragmatic sense of the word) in order to search for a means to redescribe their situations, to reattribute responsibilities, and inscribe troubles into “problematic fields.” We can think of this in terms of the micropolitics of trouble as described by Robert Emerson and Sheldon Messinger: the difficulties we face in our everyday lives happen within more or less stabilized “problematic fields,” with “typical solutions of typical problems” (Schuetz 1944). However, when these solutions are either no longer accessible or do not work, a “trouble” can progressively take the form of a problem and reassign new solutions and new “troubleshooters” (Cefai 2013). I argue that, starting from the experience of a “trouble,” the programmers came up with the idea of coding applications as new tools for the publicization of problems and for communicating with the authorities in contexts wherein the “official” channels were not perceived as working properly. Digital tools, mobile or web applications, should be considered as crystallizations of this collective inquiry:

Finally after five or six visits [to the municipal council] I decided against going there and found the website of the central city administration on the internet. I started sending them my appeals directly. Actually it turned out to be efficient because when you send letters “up there,” they trickle down to local officials, and they are so willing to get a good reputation and are so afraid of their superiors that they start working and remedy your problems. This became the basis of the mechanism that we now use in Krasivyy Peterburg. We are sending everything to the very top—to the mayor of the city—and then they send it to the local administration. (Krasimir V., founder of the Krasivyy Peterburg application)

In contrast to the image of Russian bureaucracy as a slow and inefficient machine, civic apps promise a quick and easy-to-use solution “for everyone”: the user has only to choose from a number of “categories of problems,” fill in some forms, and take a photo of the problem. The app then generates the text of a complaint using all the necessary bureaucratic boilerplate language, references to applicable laws, and sends everything to the authorities.
<table>
<thead>
<tr>
<th><strong>NAME OF THE APP</strong></th>
<th><strong>SOCIAL PROBLEM ADDRESSED</strong></th>
<th><strong>AUTHORS</strong></th>
<th><strong>DATE OF RELEASE</strong></th>
<th><strong>TECHNICAL SUPPORT</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>WebNabludatel</td>
<td>Electoral falsifications</td>
<td>A team of developers (9 persons) under the leadership of Ilya Segalovich (ex-CTO of Yandex) and a Russian NGO “Golos” specialized in electoral law. No political parties involved</td>
<td>January–February 2012</td>
<td>Mobile app (iOS, Android); and a website with an interactive map (statistics of fraud)</td>
</tr>
<tr>
<td>Krasivyy Peterburg</td>
<td>Urban problems (12 various categories)</td>
<td>A group of activists from Saint Petersburg, ex-observers of elections. Worked with several volunteer developers (2 mobile developers, 2 web developers, and a community of beta testers and helpers)</td>
<td>Autumn 2012: first idea for the app appears on a special page on Vk.com. 2013: first version of the web application. February 2014: mobile app (Android, iOS)</td>
<td>Web app (interactive problem mapping, photos of problems, user rating systems); mobile app (Android, iOS); and social network groups</td>
</tr>
<tr>
<td>RosYama</td>
<td>Potholes and road quality problems</td>
<td>Alexey Navalny (ideas and funding); Fedor E. (independent developer); Vyacheslav K. (developer of the mobile version); the developers’ team of Foundation against Corruption (Alexey Navalny’s NGO); and a community of beta testers and active users</td>
<td>2010: web app and groups on social networks (user forums)—Vk.com, Twitter. 2012: mobile app</td>
<td>Web app (interactive problem mapping, photos of problems, user rating systems), generating the text of complaint, redirecting it to the Road Inspection; mobile app (iOS, Android, Samsung Bada); and social network accounts</td>
</tr>
<tr>
<td>Project</td>
<td>Issue Description</td>
<td>First Prototype Details</td>
<td>Latest Details</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>RosZKH (formerly “Dom. Dvor. Dorogi”)</td>
<td>Problems with communal services (light, electricity, gas, in-house commodities)</td>
<td>First prototype (“Dom. Dvor. Dorogi,” or “House. Yard. Roads”) in the form of a website was developed by Dmitriy L. (lawyer and activist) with the help of volunteer developers. Second version—a lighter and more functional web application with automatic generation of the texts of complaint—was developed by the Foundation against Corruption.</td>
<td>2012: website of “Dom. Dvor. Dorogi.” Late 2012: web app by Foundation against Corruption, and pages on Vk.com (user forums) and Twitter</td>
<td></td>
</tr>
<tr>
<td>Zalivaet.spb</td>
<td>Leaks in roofs</td>
<td>Fedor G. (student in sociology, developer-amateur) built this web platform after a severe leak in his room</td>
<td>2010 Blog (used as a web application to post personal cases of leaking roofs)</td>
<td></td>
</tr>
<tr>
<td>OpenSalary</td>
<td>Inequality of teachers’ salaries and corruption in educational institutions</td>
<td>Trade union of teachers “Uchitel,” and Alexey M. (developer), with the help of “Teplitsa Sotsial’nykh Tehnologiy”</td>
<td>2012 Web application with a map that reuses and treats OpenData of the Ministry of Education and visualizes teachers’ salaries all around Russia; and Twitter</td>
<td></td>
</tr>
</tbody>
</table>
Furthermore, as mentioned by Krasimir, civic apps send the complaints to the very top of the city hierarchy. This creates a situation where city hall or the Inspection Department is obliged to redirect the complaints to the local city workers and delegate responsibilities to the corresponding services. As Fedor, the author of one of the applications, explains, “This algorithm reuses the ‘fear of superiors’ inherent to Russian administrations: when they see all these signatures and stamps, they can’t but react.”

Civic applications act as techno-juridical filters that turn personal experiences of trouble into transcoded alerts, which are formatted, translated, and expressed in a specific language. By proposing a standardized classification of problems built on legal texts, civic apps act as translating tools, adapting the language of citizens, their emotions and affects, into a language that is not only comprehensible to the city’s civil servants, but written in a form that virtually forces them to act on the complaints. Francis Chateauraynaud and Didier Torny (1999) argue that for the civil servants “it is the code itself that constructs the event,” and the app has been precisely coded to enact the politico-bureaucratic code that constructs the event.

Civic applications thus transform the practices of complaint that have been historically central to Russian culture, to the point of constituting a specific literary genre (Bogdanova 2013; Dewey and Kleimola 1970; Lambert 1985). Elena Bogdanova (2013) has claimed that the “textual space of a complaint contains the language of both sides (the author and the addressee),” but the civic apps transform this scenario into one in which the author’s language is replaced by that of the machine. The justification is thus transformed from one that mobilizes categories of personal experience to one that relies on legal and technical norms. Forging this interpersonal-technical language is also a necessary step in moving from an experience of trouble (which is inherently personal) to a tool that, instead, can be used and reused by thousands of different citizens.

The WebNabladatel app for electoral monitors illustrates the process of transformation from indignation (experience of a problem) to code. The team that worked on the application was formed ad hoc on the basis of a shared experience. They were all witnesses (either direct or through thousands of YouTube videos) of massive falsifications of the electoral process on December 4, 2011. Given the widespread awareness of the problem, and the fear of a repeat performance at the upcoming presidential elections in March 2012, the idea of a mobile application for election monitors was floated on Habrahabr.ru—the most popular forum for IT professionals in Russia.
It was Ilya Segalovich (one of the opinion leaders of RuNet at the time and the chief technology officer of Yandex, Russian’s most popular search engine) who posted the idea of an “electronic observers’ diary.” Published on December 22, 2011, this post received 117 likes and was commented on 398 times over the next forty-eight hours (Segalovich 2011). The comments included about forty stories by people who were both IT professionals and electoral observers, offering their personal testimonies of electoral fraud and police violence. The following is one example:

**USER 1:** The cops kicked me out [from the voting station] without any explanation: “first you get out, and then we’ll bring you your papers.” They wrote me: “Was interrupting the work of the electoral commission. . . . Violation of Federal Law #51.” This is certainly a lie. And I had lots of witnesses and a video recording I made.

**USER 2:** Have you complained about the fraud?

**USER 1:** Yes, sure. I complained to tec (Territorial Electoral Commission), to the court, to “Citizen observers” and “Demvybor” [NGOS specialized in electoral observations]. All complaints normally should be addressed to tec, and they have, according to the law, to treat immediately every complaint they get, as soon as they get it. I brought my first complaint to tec at noon. They told me they would examine it in one hour. Firstly, I was just naively waiting, than I split and came back to the voting station. When I came to see them next time (at 23:30 in the evening) I found out that they had not treated any complaint and any declaration at all.

(From an online discussion, posted on December 22, 2011, at 14:31)

Accounts of the inefficiency of official institutions responsible for controlling the electoral process (such as the Territorial Electoral Commission) coexisted in the same virtual space—the Habrahabr.ru forum—together with the first spontaneous technical design specifications, ideas for the application’s interface, and a checklist. The back-and-forth between stories of fraud and design solutions contributed to building connections between an experience of trouble (the physical and legal reality of electoral observations) and a “user experience” (how to put these heterogeneous accounts
into the mobile app). Just a few days after the now-famous post by Ilya Segalovich, a working team of twelve active members plus five more in charge of specific tasks such as legal consulting or beta testing was constituted. The story of WebNabludatel’s creation thus exemplifies what I call “stigmergic collaboration” (Ermoshina 2014b): It took only one month to go from the first online conversations and elaboration of ideas to the final realization in a digital interface; all coding work was performed on a volunteer basis with everyone working efficiently against a nonnegotiable March 4, 2012, deadline—the day of the presidential elections. In one month this decentralized team (everyone was working from their own place) came up with a website and a mobile app for Android and iOS.

The complexity of electoral observation practices and a multitude of possible microsituations of fraud can become obstacles for observers who have no experience or expertise. The mobile app was supposed to serve as a mobile guide to help, instruct, and prepare as many independent observers as possible. But how to put fraud into code? How to develop an interface that would take into account different cases of anomalies and illegal situations? The first step consisted of building standardized electoral “scenarios”: from the early morning, when the voting stations open, to the late night, when the votes are being counted. To better classify different kinds of fraud and develop the guide, developers worked with legal experts. The team collaborated with “Golos,” an NGO that specialized in electoral code and in the training of observers. The digital interface was based on several documents. The first was called the “Roadmap of Observers,” a guide printed on A4-size sheets of paper and distributed by NGOs before election day. It served as a base for the WebNabludatel team to prototype the “electoral scenario.” Another document was the Electoral Code of the Russian Federation, the document that was translated into the final checklist of the app. Tatyana M. explains this translation:

In the beginning we had a very big text of electoral code that should be about two hundred pages I think. So . . . I and Grigoriy M., a lawyer from “Golos,” made a kind of draft of our menu. It had to be easy—imagine, you are an observer, you come to the voting office in the morning, so . . . what should you do right from the start? You should verify that the urns are empty, if the papers are here, and so on. . . . So we made a list of questions for every step. This list was about twenty pages long I think, so I said, “It is not possible to put all this text in a mobile app. A user will just be lost in it.” So I worked till I managed to reduce these materials to only six screens with essential questions. Everything in our app is based on the law.
Indeed, Lessig’s famous motto “Code is law” becomes particularly relevant in the context of Russian civic applications: it is the language of administrative code, the official technical and legal documents that inspire developers and UX designers to build their interfaces. All four urban civic applications mapped out in table 3.1 reuse legal codes to construct their lists of problems, classifications, and check-lists. For example, in order to build the classification of anomalies and the list of categories to choose from, the developers of RosZKH relied on the text “Norms and Rules for Technical Uses of the Housing Fund”:

Actually, a fault is everything that deviates from the ideal state of things, and this ideal state is very well described in the “Norms and Rules…” For example, they specify that all metal door accessories, like door handles or door hinges, must be polished and shiny. So, anything that is not in these norms is a fault and we have the right to report it because we pay for it every month. (Dmitriy L., author of RosZKH)

Similarly, RosYama uses appropriate legal and administrative texts to define which kinds of road defects are categorized as “potholes,” that is to say, defects that the city is legally obliged to repair.

As Geoffrey Bowker and Susan Leigh Star (1998, 232) have shown, classifications and standards are important as “sites for mediation between the technical requirements of the systems developer and social and political requirements of the community.” Indeed, the categorizations and standardizations at the core of the civic apps were crucially important both technically and politically. Politically, the categorization is important as it translates indignation into an “account” (Garfinkel 1967) that can be transmitted publicly to official institutions or the press. Technically, developers need to have a set of elements to which they can attribute certain values, like “true” or “false.” While they cannot code an app “against corruption” in general, they can code a set of small tasks or questions that by answering allows a user to participate in gathering data on corruption. This is what the discourse of crowdsourcing calls “microtasking.” The translation occurs when these microtasks are calculated and aggregated by the machine and represented in the same database: every single pothole declared with the help of a mobile app becomes part of a big collaborative map of potholes. By the mechanisms of multiplication and reiteration, an individual problem is inscribed into a more global political context. That is how the struggle against potholes becomes a struggle against corruption (Ermoshina 2014a) and complaint becomes a very specific form of civic engagement.
CONCLUSION

I have sketched a portrait of Russian civic hackers—coders, computer scientists, UX designers, and geeks—who work on a very specific kind of product: civic mobile and web applications. Though these civic apps are very popular in Russia, it is difficult to speak about Russian civic hackers as a consolidated community; they exist more as a moving, nomadic, and fragile network. Its members are coding for social or common good at the margins of their day jobs as programmers, designers, and scientists. Some may quit civic hacking after having built only one project, while others may instead become “IT volunteers” with a political organization. There is no official membership in this movement; one becomes a “member” by simply hacking and making.

Russian civic hackers’ national identities equally fluctuate. On the one hand, they refer to a transnational, borderless FOSS movement, work from abroad, travel a lot, engage in exchanges and conspire with the West, the East, and Africa; but on the other hand, they share a certain form of patriotism. By coding civic software, they produce utopias and advocate situated visions of a “better Russia.” These visions are not revolutionary. They are not about overturning the social system or radically changing existing political institutions, but rather about making tools to improve the communication between citizens and authorities concerning specific grievances and points of contention. However, by doing this they produce a very particular, hybrid vision of “common good” (Hemment 2012): the voluntary coding work of civic hackers does not seek monetization but aims at improving public space and public infrastructure—not, however, in compliance with the state. While Western versions of civic apps (such as Fix My Street or Dans Ma Rue) seem to perfectly fit into a neoliberal paradigm of digital labor, where governments and corporations benefit from the efforts produced by coders and users (Eyler-Werve and Carlson 2012), Russian civic applications are not “doing the work of the government.” On the contrary, by their very design, Russian civic apps paradoxically become instruments of citizen surveillance and control over the administration. These tools are used to build political capital and are mobilized during electoral campaigns by anti-Putin opposition.

Russian civic hacking is not about breaking the law but about automating respect for it through the translation of legal texts into programming code. Thanks to the performativity of computer code, every user gets access to legal codes and to the ability to mobilize them without having to study or necessarily understand the law. Russian civic hacking is about fixing the
system by using its own rules and mechanisms (the pipelines and labyrinths of the notorious “Russian bureaucracy”) to make it effective against its own will. It is about the domestication of the Leviathan by means of translating it into many short lines of code and into an easy-to-use interface.

Engagement through an application—“apptivism,” as I call it—is a ubiquitous form of political participation. The act of participating is intermediated by the interface and becomes part of day-to-day life. Russian civic applications rethink the communication between citizens and authorities, and in doing so they also rethink the relation between global and local. That a number of civic software projects may die out or prove to be ineffective does not detract from the importance and success of civic hacking as a movement—one that provides alternatives to traditional repertoires of political action (such as street activism, demonstrations, strikes, or petitions, which are either inefficient or forbidden in Russia).

Apptivism enables the innovation of Russian political practices and forms of participation as much as it contributes to the innovation of Russian IT. The interfacing of these two halves of the new “public sphere” relies in part on the work of a new kind of intermediary institution that seems specific to the Russian context, as they tend to compensate for the lack of infrastructure for innovation, already mentioned by several authors in this volume. One example is Teplitsa Sotsial’nykh Tekhnologiy, which identifies its mission as “building bridges between IT-professionals and NGO workers” (Teplitsa, n.d.). In fact, this organization sustains the civic hacking movement by organizing webinars, hackathons, workshops, and other kinds of meetings that promote the use of code to solve actual problems in Russian society. They have also institutionalized the status of the civic hacker with their recent project, “IT-volunteer”—a platform that unites IT professionals willing to collaborate for free on specific tasks proposed by NGOs. This NGO tries to compensate for the absence of conditions for social innovation in Russia and enjoys relative freedom of action because of the lack of governmental control in this field. Neither purely political nor solely technical or commercial, the civic hacking scene constitutes a unique case, even within the scope of this volume: it requires almost no relations with either academia or governmental agencies and is deployed on the margins, in between different fields.

I hope that taken together these examples demonstrate the remarkable, interesting, and emergent qualities of the civic hacking movement and its constitutive connection to both the technical expertise of Russian coders and the specificity of the Russian sociopolitical context. It is literally about code and country.
1. According to a study conducted by Alexey Sidorenko, the founder of Teplitsa Sotsial’nykh Tekhnologiy (the Greenhouse of Social Technologies), a Russian nongovernmental organization (ngo) specializing in the promotion of civic applications and computer literacy among social activists.


4. The reaction of authorities to civic software projects is uneven and varies according to city and region. Thus, the city of Moscow—with its trend toward “modernization” and “e-government” launched by Dmitry Medvedev during his presidency—has agreed to collaborate with the project RosYama, an app that generates alerts about potholes. The Road Inspection of Moscow integrated user-generated data from RosYama into their platform. A counterexample is the city of Saint Petersburg that refused to accept alerts sent through the RosZKH application, designed to generate complaints about problems with housing utilities.

5. All interviews quoted in this chapter were conducted in Russian and translated by the author, except for the interview with Emily Parker, conducted in English by the author.

6. Alexey Navalny is a famous Russian blogger, lawyer, and entrepreneur. He is the conceptual author of the RosYama and RosPil apps, and also the founder of the Foundation against Corruption, an ngo that hosts several projects (RosYama, RosPil, RosZKH, Dacha.fbkinf0). Navalny had a large office in Moscow where I could observe their work on different apps over the period of a few days.

REFERENCES


Bennett, Elizabeth A., Alissa Cordner, Peter Taylor Klein, Stephanie Savell, and Gianpaolo Baiocchi. 2013. “Disavowing Politics: Civic Engagement in an Era of


This page intentionally left blank