entirely, yet all of those concerns are valid and of equal importance.[‡] Much like the rest of the industry, Free Software in general and KDE in particular are coming to terms with pervasive concurrency and increasingly distributed processing. Users rightfully demand easier-to-use, cleaner, and more beautiful interfaces; responsive, well-thought-out, and not overly complex interactions with software; and high reliability, stability, and data safety. Developers expect language-neutral extension points, well-maintained APIs, binary and source compatibility, clean migration paths, and functionality on par with the commercial offerings they are frequently accustomed to from their day jobs.

But even more daunting are the social and coordinative aspects of a larger Free Software project. Communication is key, and it is hampered by time zones, cultural barriers, more or less reflected preferences and prejudices, and also simply by physical distance. Discussions that can be finished in 15 minutes in a stand-up office meeting may take days of arguing on mailing lists to allow the opinions of team members from the other hemisphere to be heard. The process of reaching consensus often is at least as important as the resulting decision itself, to keep the cohesion of the group intact. Naturally, more patience, understanding, and rhetoric wit is necessary to gain a favorable consensus. This is much to ask from people who aced math and physics classes without flinching, but avoid getting a haircut because they do not like the talking to nonprogrammers that such an activity entails. With this in mind, the amicable spirit of the annual Akademy conferences is a wonderful experience (see the next section). We have never seen such a massively diverse group of old and young people from all over the globe, from rivaling countries, countless nations, seemingly different worlds even, and with all shades of skin set aside differences to argue—heatedly, but rationally—about arcane C++ subtleties.

And then, aside from technical and personal aspects, there is a third major influence on how a Free Software project fares: structure. In some ways, structure is inevitable for Free Software groups. Larger and better-known software projects especially need to hold trademarks, receive donations, and organize conferences. The time when this is possible solely through the good will of spouses and weekend trips is usually already over when a project is mentioned publicly for the first time. In some respects, structure (or the lack thereof) determines the fate of the community. An important decision has to be made—whether to go corporately dull or geekily chaotic. The answer is not trivial, because multiple trade-offs are involved: funding versus freedom from external influence; stability of the development process versus attracting the most brilliant cave dwellers;[§] visibility and mind-share versus focus on impressive technical results. Beyond the point where structure is a bare necessity, there are options for the levels of bureaucracy. Some projects do not even have a board of directors, whereas some have

[‡] This is fundamentally different from software that is entirely produced for a market. Free Software can cater to the needs of the blind, for example, without having to justify the considerable effort needed for that with corresponding sales figures and expected returns on the investment. That is one of the reasons why KDE is available in so many more languages than proprietary competitors, for example.

[§] We use that term with a lot of affection. It cannot be denied, though, that many of our dearest friends could do with a tad more exposure to sunlight and healthy food, overall.