

# on the integration of interaction architecture in open-source projects

**peter siking**  
principal interaction architect  
man + machine interface works  
fehrbelliner straÙe 85  
10119 berlin N-54  
[peter@mimiworks.net](mailto:peter@mimiworks.net)  
[mimiworks.net/blog](http://mimiworks.net/blog)

## abstract

Working as an interaction professional on an open-source project is both a very familiar experience and an opportunity for innovation. It also means working on a reference project that can be publicly shared and publicised about.

Contribution is the key to acceptance in open-source and only in close co-operation with developers we are able to introduce good user interaction in open-source software

## Keywords

interaction architecture, open-source, integration, innovation

### 1.0 what's so different about open-source, anyway?

Less than you would think. Having worked in the commercial software world since 1993, and increasingly involved with the open-source world for the last few years, I see a 90–10 rule here: 90% is the same, 10% is different between these two worlds.

#### 1.1 same old song

Between the two worlds I encounter the same spread in project size; the same spread in management, from headless chickens to bureaucratic; the same tendency towards working without any design documentation; the same growing pains, where a couple of guys knowing every detail by heart grow into dozens, with only the same couple of guys having the crucial details in their heads; the same software entropy, where nimble, high performance software architectures get bloated and maintained into fragile spaghetti code.

These similarities make that experience gained by interaction professionals while working on open-source can be used in the commercial software world and vice versa.

#### 1.2 more, well, open

Discussions, work in progress, design and specifications, resulting software and the opinion of users, they are all there, out in the open on the internet, for open-source projects.

I see this as an excellent opportunity for usability and interaction professionals to evangelise our trade, to achieve higher acceptance in the software engineering world. And it is also a chance to work on an inspiring real-life reference project, which for once is not under an NDA.

Openness also extends to open-source projects giving carte blanche to usability professionals to unleash any kind of (brand new) method of their choice on any number of users, again without the interference of an NDA.

Also there is a continuous release of  $\alpha$ - and  $\beta$ -releases, that are not only freely downloadable, but also **get used** by sizable, enthusiastic communities in real-life situations.

This gives interaction professionals the chance to release experimental interaction into the real world, and gives usability professionals the chance to

test these experiments under real-life conditions.

Combining all this, I see that the open-source world could be a hotbed of innovation in both user interaction and in usability methods, if professionals in both fields would make use of what's so different about open-source.

### 2.0 integrating in open-source projects

The power to change software in the open-source world is based on contributing code. The more code one contributes, the more influence one gains in the project. Interaction professionals cannot try to write code, because there would be not time for user interaction work. So we have to achieve our goals in other ways.

Forgetting about code, the key is to contribute to the project. Straight at the beginning of the cooperation we can make sure that the methods we use to obtain scope and purpose from the developers also reward the open-source project with a new sense of direction and identity. I do this with my product vision method.

### 2.1 parallel society

When interaction professionals start working on a project they are in effect starting up a parallel project to the development counterpart. This has to run in parallel until we have mastered an overview of the functionality and have a series of solution models in place for every aspect of the software. The best we can do until that moment is to work as openly as possible, to create a sense of progress and a sense of professionalism that goes into our work.

I do this by working together with my team in a public wiki, where anybody interested can see us working (but not edit). I also blog about the projects to explain our progress.

### 2.2 working together

Once we are on top of the situation, the only way to get good user interaction into code is to contribute by working very closely with the developer(s). Also writing UI specifications for the particular piece of software that is being (re)developed is not only a tangible contribution, but it also transfers a more complete picture of the solution to the developers.

Another way to contribute for interaction professionals is to help with bug triage for everything UI related in the project's bug tracker. Demonstrating our competence in these matters and saving the developers work can only help with integration.

### 3.0 from evaluation to redesign

With the expert evaluation done and all the analysis written up in the public wiki, the interaction professionals can contribute again by prioritising all major UI issues and deficits. This is the start of professionals setting the agenda for user interaction. Combining our agenda with both the technical plans and sentimental arguments from the developers dominates the roadmap discussion.

This roadmap allows us to work out and specify the user interaction solutions before we start working with the developer(s) to implement them.

# ISO und Usability