**Hackerspace Design Patterns 2.0**

**[Intro]**

**[Our Community]**

I so love it here at Camp. Last night, as I was hanging out on my computer, writing this talk, I was distracted by way too many interesting conversations.

* A Pirate Party employee talking about copyright
* an acquaintance who’s used the last year to create his own trippy game
* how to encode and decode transmissions with our lovely Rad1o badges
* whether the funding sources for our projects matter
* how to promote open science
* secure password managers
* filmmaking
* and, the merits of sleep…

This is our community.

Though my first hacker conference was in 2006, the turning point for me was 8 years ago – at CCCamp2007.

That’s when several of us from North America came together to learn about hackerspaces here in Germany. We wanted to make hackerspaces happen in our home towns! 3 German hackers agreed to put together a presentation about how to start and run hackerspaces. They researched what worked well, and not so well, at the hackerspaces of the time in Germany.

**[Hackerspace Design Patterns 1.0]**

They presented their research at 24C3 in the form of patterns: The Hackerspace Design Patterns. Within several months, there were a few hackerspaces in North America, including Noisebridge, the hackerspace in San Francisco that I co-founded. And a year later, there were over 100.

**[2,000+ Hackerspaces and Growing]**

There are now over 2,000 hackerspaces listed on hackerspaces.org, a wiki website that exists for hackerspace communities to help each other, which was also inspired by the Hackerspace Design Patterns talk.

The Hackerspace Design Patterns started the hackerspace movement!

**[My first Hacker Conference]**

When I went to my first hacker conference in 2006, I’d never even heard of hackerspaces. I’d never even heard of hacker conferences before. I was kinda nervous. I didn’t know what to expect. As it turned out, the conference was full of introverted geeks! And, it was the first time I felt wonderful, and at home, in group. This was because of the community there. It was a community of wonderful geeks!

**[Community]**

Community is important. For all of us. We need it. (This is why we’re here, right?) As a species, we no longer need community to survive (as we used to). But we need community to feel like our lives matter. To feel part of something bigger than ourselves. As well as cool places with blinky lights and some nice tools, hackerspaces provide us community. I think it is perhaps the most important thing they provide us.

But, as it turns out: community is hard work!

The Hackerspace Design Patterns are really a great set of ideas for forming communities that work well, and keeping them working well.

Even though they were written and presented 8 years ago, with only a small number of hackerspaces at the time to draw from, they hold up remarkably well, and are still very relevant today.

I don’t have time to go over all of the original Hackerspace Design Patterns, but I’d like to go over a few of them, and give you a little taste.

**[The Infrastructure Pattern]**

**The Infrastructure Pattern**

**Problem**

You have a chicken-and-egg-problem: What should come first? Infrastructure or projects?

**Implementation**

Make everything infrastructure-driven. Rooms, power, servers, connectivity, and other facilities come first. Once you have that, people will come up with the most amazing projects you didn’t think about in the first place.

**[The Grace Hopper Pattern]**

**The Grace Hopper Pattern**

**Problem**

Is now really the time to start your hacker space? Shouldn’t you wait? Have you really thought of all the problems?

**Implementation**

Sure it is the time!

*It’s always easier to ask forgiveness than it is to get permission.*

(Grace Hopper, US Navy Rear Admiral and computer scientist)

It’s important to start. Many problems you think of before will vanish as soon as you get started. When in doubt, do it!

Outgrowths of the above that have been generally useful rules of thumb at hackerspaces around the world:

Don’t solve problems that haven’t happened yet.

Do-ocracy. You don’t have to be good at something to do it. You get good at it by doing it.

**[The Tuesday Pattern]**

There is a Plenum Pattern, which expresses the importance of regularly scheduled weekly meetings where people can come together to make decisions and solve problems.

The next Pattern that follows the Plenum Pattern is:

**The Tuesday Pattern**

**Problem**

Every weekday sucks. You will not find any day when every hacker can attend a meeting. Someone always has an appointment.

**Implementation**

Meet on Tuesday. Since all days are equally bad, just pick the Tuesday. End of discussion.

At the beginning of Noisebridgde, before we had a space, we had meetings every Tuesday with up to 50 people attending. But, Tuesday was super-problematic for people. So, for 3 weeks, we made it a priority to choose a procedure to pick a better day of the week. We finally chose a method where everyone gave their #1, #2, and #3 best days of the week, and came up with an algorithm for compiling the data, and coming up with the absolute best day of the week for us to meet. Which, turned out to be: Tuesday.

**[The Roommate Anti-Pattern]**

There are also Anti-Patterns.

**The Roommate Anti-Pattern**

**Problem**

You need a space for meetings and as a lab, to store and work on materials for projects. In order to minimize rent or out of sympathy, you think it’s great when someone lives in your space. But

somehow it doesn’t work, as you cannot use the lab anymore.

**Implementation**

Guest are fine, but don’t let anyone live there. Kick them out if necessary.

At Noisebridge, and many other spaces around the world, people attempting to live in the space became a serious problem.

**[Not a Cookbook]**

The idea of Patterns is that they aren’t rules to follow. They are patterns of what has worked well, and not so well, at many places. The idea is not to do things the same as other spaces. But, it is certainly useful to learn from others’ experiences. And, if you do something different from the Patterns at your space, do so consciously.

For instance:

/tmp/lab/ started in a squat in a not-very-nice suburb of Paris. They chose to have someone live there to have a living presence in the space 24-hours a day.

Also, for instance:

**[The Kitchen Pattern]**

**The Kitchen Pattern**

**Problem**

As a human being, you need food. As a hacker, you need caffeine and food at odd times.

**Implementation**

Have a kitchen at your space. Nothing brings people together like cooking together. Have fridges for Club-Mate. Selling soft-drinks will help you raise money for the rent. Invest in the single most

important piece of hardware: a dishwasher. Have a freezer for pizzas and buy decent kitchen equipment. Show nerds how to cook real food.

**[The Kitchen Anti-Pattern]**

I’ve been to so many hackerspaces where eating together with people was wonderful. And I’ve enjoyed Noisebridge’s kitchen for the first years of our existence. But, as our community grew, the kitchen became problematic, more problematic, and eventually disruptive – where not nearly enough people cleaned after themselves, attracting cockroaches and rats from the grocery downstairs, and attracting people who came only for the kitchen, and not our community or our vision. We ended up getting rid of our kitchen last year, since it became an Anti-Pattern.

**[More!]**

That gives a taste of the original Hackerspaces Design Patterns. There are many other Patterns, covering topics such as sustainability, funding, and conflict resolution.

So, now, with so many more hackerspaces in the world, and with 8 more years of collective experience, I’d love to add to the Patterns for the benefit of the hackerspaces into the future.

Hard to say for sure, but I’ve been to about 250 hackerspaces, or so. But, rather than me decide upon new Patterns, I put out the word on various hackerspace email lists, as well as the hackerspaces.org email list.

And, the preliminary results are:

81 pages of raw notes. With more coming continuing to come in daily. Clearly people at hackerspaces around the world are thinking of this! We think it is important and very useful.

I can’t even begin to give a summary of all the raw data I have. But, I’ll give you a taste of what I have.

The notes I’ve compiled from others cover such topics as:

* funding
* people who are toxic to a community
* keeping people motivated
* group organization
* conflict resolution
* how to deal with stuff
* and many other important topics.

**[The Mentor Pattern]**

I thought this was an interesting and helpful one, perhaps based on the mentor program recently implemented at Chaos Congresses:

**The Mentor Pattern**

**Problem**

On-boarding new members is difficult. New members can have trouble finding the resources they need and learning their responsibilities as members.

**Implementation**

Every new member is assigned an experienced member as a mentor. When the member has a question, they can go to the mentor to find the answer. If the member is acting out of line with the group's principles, the mentor can talk to them.

**[The Cultural Immune System Pattern]**

This Pattern is if from Noisebridge, gained from some pretty bad experiences, that nearly led to us dissolving a year ago (more on this coming up):

**Cultural Immune System Pattern**

**Problem**

You want people in your hackerspace who are good for your community, and want to keep them happy and safe -- and you don't want to attract people who will hurt your community or drive away people who are good for your community.

**Implementation**

"Cultural immune system". Get the entire community involved with attracting new people who belong. Greet new people at the door, give them a tour, including what your community is and is not about. Keep people who don’t belong out of the space. Define what it means to belong and not belong.

At Noisebridge, someone who belongs: a person who both benefits from, and contributes to, the community. “Contribute” can take many forms: financial, makes people feel good, helps clean, helps maintain, teaches classes, shares their skills and knowledge, etc.

**[The Reboot Pattern]**

**Reboot Pattern**

**Problem**

At the beginning of a space, most people in the community are actively involved on working together in creating the space. This is very bonding. Over some time, with people coming and going, this feeling fades. Newer people don’t feel as much a part of the space. Older people may feel burned out. There may be less of a sense of group cohesion. Stuff stops getting done, and infrastructure deteriorates. People who don’t belong at the space may start becoming more numerous and problematic.

**Implementation**

Reboot the space!

When your computer starts acting badly, it often helps to turn it off and on again.

Close the space, kick everyone out, and only let people in for some period of time if they are helping fix the place up. This renews the sense of everyone creating the shared space, and is very bonding. Start the reopening with a nice party. After opening, continue to keep out people who don’t belong, making the space more attractive for those who do.

This was so successful at Noisebridge a year ago – like, super-amazing-wow-successful—that we decided to have a mini-Reboot annually. We just had it this year, and are having our Renewal Party later today.

You don’t need let your space get into a bad state before doing a Reboot. I think having a Reboot is a great thing – have one annually!

An alternative to a Reboot, is to create a big project that engages a large number of people in the community – something that people really enjoy, and that has a big impact.

**[The Noisebridge Reboot -- before]**

**[The Noisebridge Reboot -- during]**

**[The Noisebridge Reboot -- after]**

**[MORE!]**

The above examples are a very small percentage of the suggestions sent to me for consideration of updating the Hackerspace Design Patterns.

I hope that these serve as an opening for further discussion.

There is great diversity in the hackerspaces that exist in the world. Yet, I think there are still Hackerspace Design Patterns that could be distilled out of our collective experience that are universally useful for all hackerspaces to ponder. And, so it would be way useful to compile such an updated Hackerspace Design Patterns 2.0

I also think it would really useful to compile Hackerspace Design Patterns 2.0 that are useful for subsets of hackerspaces, such as ones for urban spaces, or ones in developing countries.

The world needs hackerspaces! There are 7 billion people on the planet, and only a small number of us have access to the opportunities they offer. Most of us here today are all lucky enough to have access to these incredible communities. But, clearly we need more hackerspaces on planet Earth! Let’s make more hackerspaces! And let’s empower and help others to do so.

Compiling Hackerspace Design Patterns 2.0 to add to the original Patterns is one helpful step.

**[An Invitation]**

Discussion / Workshop for Hackerspace Design Patterns 2.0

Now

at NOPE

11:15am to 1:30pm

**[Another Invitation]**

Also:

Discussion / Workshop for Revitalizing the hackerspaces.org website

Sunday (tomorrow)

at NOPE

7pm – 9pm