

Starting an Emacs Meetup

Harry R. Schwartz

September 14, 2015

{{ “Q7p69ZVs5CA” | youtube }}

I recently gave a talk at EmacsConf 2015 on starting meetups, so if you’d rather read this article as a series of bullet-points, you can flip through my slides.

My goal here is to walk you through the steps of starting an Emacs meetup for yourself. Most of this information should also apply to any free-software-related meetup, and, more broadly, to meetups in general.

Introduction

I’m Harry Schwartz, a developer at thoughtbot and happy Emacs user. thoughtbot is a terrific company, but it’s fond of vim—in fact, in the whole company, only Eric Collins and I are full-time Emacs users. We developed an (almost entirely) unwarranted persecution complex and decided to start an Emacs meetup as an immature act of rebellion.

We launched EmacsNYC last spring (that’s 2014) and it’s been happily growing ever since. We arrange monthly meetings, corral speakers, and film our talks and host them online. We’ve now got almost 300 members, which makes us the largest Emacs meetup in the world (that we know of, anyway).

I’ve since moved to Cambridge, MA. EmacsNYC is still running just fine under Eric and a new co-organizer we recruited from the community (the excellent Mr. Zachary Kanfer). And, if I get my ducks in a row, we should soon have an EmacsBoston!

So! Let’s talk strategy.

Find a cofounder

Running a meetup is a lot of work. Well, that’s not *entirely* true— it’s usually only a few hours each month—but during those hours you’ll really want more than one pair of hands. You can technically do it alone, but you’ll be much better off if you can find yourself a cofounder. A cofounder will be able to help you with logistics, finding speakers, setting up equipment, meeting delivery folks, and all the other tiny chores that are involved in running a meetup.

They can also help out by providing emotional support. Even if you find that you're doing most of the work yourself, they can still be an ally and a cheerleader; it's entirely possible that everyone thinks you're a bit odd for organizing an Emacs meetup, so if you have a friend who *gets it* you'll have a much easier time dealing with the hassles.

Eric was terrific as my co-organizer for EmacsNYC. We actually ran two meetups together (see also, Ruby Project Night NYC). I mostly focused on the Emacs meetup, and he mostly focused on the Ruby one, but we always collaborated on planning and organization and generally kept each other in the loop. Super helpful!

Find a sponsor

You need space, food, various online services, and possibly recording equipment and video production. None of those things are free. You absolutely need a sponsor.

Our company, thoughtbot, sponsored EmacsNYC. They provided us with space (originally through a coworking space, while we were still nomadic, and now at our permanent office in NYC), paid for our food and hosting, let us use their recording equipment, and our company's producer even edited our videos for us (and did a great job).

We were extremely lucky—it's not always easy to find a good sponsor.

If you're working in software, ask your company first. A lot of companies like being associated with meetups (even odd ones) since it gets their name out in the community and helps them make connections, hire developers, and (if they do consulting) find clients.

You might be willing to pay for hosting and domains yourself, but you'll definitely want someone to pay for food. Every attendee will eat at least a few dollars worth of food per meetup, so unless you're *really* willing to drop some money you'll want a corporate sponsor.

Space is a bit easier. Most cities have a few software companies with event spaces that they're willing to loan out to the community; you've probably been to a few if you regularly attend other meetups. Universities (especially ones with well-established computer science departments) are sometimes willing to host community events. Many cities also have a hackerspace or two, which are generally pretty welcoming toward free-software-related meetups.

Things to look for in a space

Consistency is great for a meetup. It feels much more *real* if you know it'll happen every month on a certain day at a certain location, and it lets attendees know that they can go ahead and mark it on their calendar with confidence.

This means that it's important that you get find a meetup space that you can rely on. Thoughtbot worked out of a coworking space for the first year or so of EmacsNYC. The company that owned the space was happy to host our meetup, but every month we'd be put in a different building in a different neighborhood in Manhattan. That gets a little old: attendees get lost, it's hard to plan for food delivery, and every building has a different security procedure.

Some of the questions you should investigate when looking for a space:

- Make sure people can get in after hours. Is anyone manning the door? Can you post signs?
- Is there a doorman? Does he need a list of names? Can you *get* a list of legal names from your attendees?
- Are there enough chairs?
- Are the bathrooms locked after a certain time? (This is sometimes the case, and it is *frustrating*)
- Where will the food go?
- Is it a shared (occupied) space? If so, will other people mooch your food? Is that a problem?
- Will the space be quiet enough? Is there, for example, muzak that can't be turned off?
- Is there a projector? If not, can you bring one? How will the speaker plug into the projector? Do you have a spare VGA/DVI/HDMI cable?
- Is there somewhere to put a camera?
- Will the speaker have a podium, if they need one?
- Will there be enough light on the speaker for recording?

Feed the people

If people are skipping dinner for a meetup, they usually expect to be fed. You must feed them.

EmacsNYC has generally taken the path of least resistance here and ordered a bunch of pizzas (assume about one large pizza per four attendees). We generally also provide drinks (water and beer). This is very easy and rarely causes any hassle; it's what most groups do, so everyone just kinda expects it.

I don't really think that only ordering pizza is a good thing, and I'm planning on doing it differently for EmacsBoston. Here's why:

- It's just unhealthy. By setting up a pizza-only situation you're kinda pressuring people to eat badly.
- Not everyone feels comfortable around a bunch of beer-drinking dudes.
- It's kind of inherently exclusionary. If you're a vegan, or have a gluten allergy, you're probably not too excited about pizza. We tried vegan-and-gluten-free pizza, and that's definitely better than nothing, but it really feels like an afterthought. It's not fun when your meal is just "the main dish without the good parts."

- This is just correlation, but I've noticed that women-friendly meetups generally don't serve pizza, and when they do they usually provide an alternative or two. Attendees seem to like having a healthier option.

There are tons of catering options available these days, and it's not *that* hard to organize a meal that isn't just pizza.

So! All that said, consider the following:

- Your food, whatever it is, should be set to arrive about half-an-hour before you expect people to show up. The delivery folks might be late and it'll take time to set up.
- People will go for seconds during the talk. Set up the room such that this isn't too disturbing (e.g., no one should have to walk in front of the camera or projector).
- Have some extra napkins and disposable cups (and, ideally, cutlery) on hand. Caterers and delivery folks sometimes forget something. Alleviate this by stockpiling.

Finding speakers

We haven't had too much trouble finding speakers for EmacsNYC. Generally people in the community have been happy to volunteer (usually with only the slightest of pressuring :-).

Occasionally, though, we can't find a speaker, or our arranged speaker has to drop out at the last minute. That's not a catastrophe. We'll still order food, and we'll still have the meetup, but we'll just socialize and chat about what we've been up to lately. We might have an impromptu lightning talk or two.

It's important to still have the meetup, even if you can't get a speaker. We're trying to build a community here, and a consistent schedule helps with that.

Coming up with topics

Meetup.com lets you automatically ask new attendees a few questions when they join your group. We asked "What's your experience level like with Emacs?" and "Would you be interested in giving a talk?"

Several people replied with something like "about thirty years" and "sure, but I don't know what I'd talk about."

That was infuriating! If you've been using an editor for *thirty years* I'm pretty sure I'd love to hear just about anything you have to say about it.

It's really common for people to be interested in speaking but not to have any ideas for a topic. They assume that if *they* know it, then *everyone* must. To help fix that for EmacsNYC, we wrote up a list of topics that we'd love to see someone talk about. If someone's interested in speaking, but doesn't have a specific topic in mind, we can point them to that to give them some ideas.

By the way, your speaker doesn't have to talk about Emacs. At the end of 2014 we had a Holiday (Key-signing) Party, where George Brocklehurst gave a terrific talk on PGP. We're currently lining up a talk on building mechanical keyboards. It's fun to cover unrelated topics now and then, so long as they're likely to interest most of the group.

Setting your speakers up for success

Once you've got a speaker, there are plenty of things you can do to help make sure they succeed.

- Get a title and abstract from them as early as you can. This'll give you some time to announce the meetup, give the attendees a chance to mark the meetup on their calendars, and it'll force your speaker to start planning out the scope of their talk
- Prepare a speaker's guide with some tips and advice.
- Offer to help them rehearse. If you've both got the time and the space, offer to let them practice their talk with you. Doing a dry run can help them consolidate their ideas and give them some confidence.
- If you plan on recording their screen and using that in your video (if you're making one), make *sure* they install any screencasting software well beforehand. It's a huge pain to try to set this up just before they go on stage.
- Give them your number for last-minute emergencies!

Community Building

Virtually every programming community has an acknowledged problem with diversity. We're mostly beardy white dudes. And while there's nothing intrinsically wrong with being a beardy white dude, it'd be nice if we had a community where *everyone* felt comfortable.

The Emacs community has an extra dimension to the inclusiveness problem: we have a bunch of free-software enthusiasts among us, and we should try to make them included, too.

Inclusiveness

There are plenty of things we can do to make everyone (beardy or not) feel comfortable in our community. Here are a few:

Have a code of conduct

A code of conduct gives attendees some confidence that you care about their well-being and will actively work to ensure it. *Not* having a code of conduct sends the opposite message.

It takes very little work to put together a good code of conduct. PyLadies has a terrific one, and they encourage other groups to fork it (which we did for

EmacsNYC).

Follow the Recurse Center (née Hacker School) Social Rules

The Recurse Center has a great set of rules of behavior. Their description of the rules goes into more depth, but the summary is:

- No feigning surprise (“You don’t know C-h?!”)
- No “well-actuallys”
- No back-seat driving
- No subtle “-isms”

These are great rules for technical people to follow.

Provide a variety of foods

I went into this is some detail above. In short, provide vegetarian/vegan/gluten-free options, and ideally try to keep at least some of your food reasonably healthy (i.e., not just pizza and beer). We haven’t been very good about this at EmacsNYC, but I’d like to be better in the future.

Don’t accidentally discriminate by experience

There are a few folks at every Emacs meetup that have been using it since the 1980s, and they’re generally awesome. It’s important, though, to make people that are still getting started feel comfortable, too. If a newbie comes to visit, and everyone they speak to immediately tells them that they’re been using Emacs since it ran on a VAX, that newbie’s probably not coming back.

There’s no great way to counter this (the more experienced folks aren’t wrong to be experienced, though of course they shouldn’t be jerks about it) but it’s important to emphasize periodically that *everyone*’s still learning new tricks, even if you’ve been learning for decades, and that we’re all beginners at something.

Evangelize

There are a lot of people out there that might enjoy attending an Emacs meetup (well, not a *lot*, but plenty). Most of them aren’t actively scouring the web for Emacs meetups in their area, and the ones that are tend to fit our stereotype a bit. If we want more diverse people at our meetups, we need to seek them out. Go to other meetups and talk to people! Don’t be a bother, of course, and don’t detract from what that meetup’s doing, but respectfully announce your meetup’s presence and extend a welcome.

Free software folks

I consider myself to be a member of the free software community: I use free tools whenever I can, I evangelize, and I’m a card-carrying member of the FSF (note that I haven’t said “open-source” in this essay yet). That said, I sometimes make exceptions, and there are definitely plenty of people that are much more passionate about free software than I am.

Working with free software folks feels a bit like cooking Thanksgiving for vegans: we should definitely provide plenty of vegan options, but we should acknowledge that a lot of folks are gonna want to eat turkey.

In the same way, I've been trying to provide options for our meetup. Rather than use Meetup or YouTube? We've got a JS-free website with RSS and our videos are transcoded to WebM. Most people *do* want to use the non-free option, since it's generally easy and convenient, but for those who don't, there's a free alternative available.

Free software folks usually care about licensing. There are a lot of subtleties involved in choosing licenses. We've chosen to GPL our software and release our other material as CC BY-SA, and we haven't heard any concerns about that. There's no downside to licensing things correctly (you were releasing it anyway, right?), and a lot of people really appreciate it.

And finally, for heaven's sake, don't make *fun* of folks' ethical choices. I basically never hear it from Emacs people, but parts of the web development community (for example) seem to have a certain arrogant ignorance about the free software world. Don't be a part of that.

Technical groundwork

This is the least important part of running a meetup. You can run a successful meetup with almost no technical infrastructure at all. Unfortunately, our (i.e., programmers') minds are wired wrong, so we usually tend to devote the *most* energy toward that. I'm no exception, so here we go.

Meetup.com

Meetup isn't free (it's neither libre nor gratis). It costs a fair amount of money (the unlimited plan is something like \$15/month). However, if you *don't* register on Meetup you'll almost certainly have very few attendees. If you want to reach a wide, meetup-attending technical audience, Meetup is your best bet. It's basically the only option, so I'd advise you to make your peace with it and set up an account.

(Of course, yes, it's possible to run a meetup without Meetup. But you're *really* swimming upstream, and by relying entirely on a blog and IRC/Usenet/etc you'll probably be reaching only the grayest of the graybeards—and not that those folks aren't awesome, but you're probably going to have a big diversity problem *anyway*, and there's no need to exacerbate it.)

Websites

Some folks refuse to use Meetup, and that's totally fine; we can accommodate them by running our own website and cross-posting our events to both sites.

This isn't too hard. Your technical constraints are:

- It should provide a feed that people can subscribe to (RSS/Atom).
- It shouldn't use any JavaScript, especially for analytics. People don't want to be tracked.
- Ideally, it should look OK in `eww`!

An additional benefit of using a blog is that you can make your videos (if you record any) available as posts. This makes it easy for people to link to them and is generally nice for promoting your group.

You might quibble that a static site doesn't let people RSVP, so you might not get an accurate headcount (which is important for ordering food and such). Rest assured that the headcount you're getting from Meetup won't be accurate either (people will flake, or bring five friends), so this won't really make a big difference with respect to unpredictability.

I chose to build the EmacsNYC site with Jekyll, write it in Markdown, and host it on GitHub Pages. I made those choices because my personal blog uses Jekyll, I write a lot of Markdown for work, and I was planning to use GitHub to host the repos anyway.

If you're the sort of person that's excited about starting an Emacs meetup, you're probably salivating at the thought of building the website in Org-mode. That's a perfectly good choice—pick whichever tool lets you get the job done!

Regarding GitHub and GitHub Pages: I'm not *thrilled* that we use it (see Benjamin Mako Hill's *Free Software Needs Free Tools*), but I haven't pulled together the wherewithal to move us to something else. All our work is managed in git, and Jekyll compiles a static site, so if we decide to move in the future it shouldn't be difficult. I'm not too concerned about it, yet, but I might make a different choice for EmacsBoston.

Our site's source is available. Feel free to take it and tweak it!

Recording video

We record most of our talks at EmacsNYC. We capture three streams of input:

- Video of the speaker. The video camera is pointed at the speaker, not at the projector screen.
- Audio of the speaker, with a separate audio recorder mounted next to the speaker. This ensures that we don't have too much crowd noise.
- The speaker's laptop screen, using screen recording software. The quality of this is *much* better than using the camera to record the projector screen.

Once we've recorded these three files, we pass them along to our producer, who cleans them up and stitches them together into a single file.

Our logistics script has some tips about getting a good recording.

We send our speakers' guide to every speaker once we've scheduled their talk. This includes some information about setting up screen recording.

By far the most difficult part of this process have been to ensure that the speakers have installed and tested the screen recording software *before* they arrive at the event. I haven't yet figured out how to effectively encourage them to do this. I'm open to suggestions.

Hosting videos

You have a number of hosting options. We post ours to YouTube, since that's what the majority of people prefer, but any service that can stream your videos (and will pay for the bandwidth) would work just as well. Vimeo is fine, too. I don't know anything about GNU's MediaGoblin, but it might be worth checking out.

We also transcode our videos to a free format (we use WebM, but Theora is also just fine) and host them on our website (technically we're hosting them on Amazon S3 and linking to those files on our website).

Streaming live video

I've never done this, but it's definitely possible. The fine folks at LibrePlanet 2015 did a great job, and they did it entirely with free software. They've published some notes.

I've heard that the Chaos Computer Club in Germany has a terrific live-streaming guide. Unfortunately, I can't seem to find it. If you can, please let me know where it is!

Sacha Chua recently wrote a nice post about how EmacsConf 2015 managed streaming, too.

There are also a ton of gratis-but-not-libre streaming options (periscope and twitch seem to be popular). I'd try to avoid them, if you reasonably can, but they're there if you need them.

If you're going to do live video, you might as well make online participation as seamless as possible (the organizers did a great job of this at EmacsConf 2015). Setting up an IRC channel for your meetup isn't a bad way to do it. Be sure that you make online viewers feel included, too: when they ask questions, ask the speaker!

Antipatterns

There are a few things that people consistently try to do that are wrong.

Trying to use Emacs for everything

You're starting a community, and your goal should be to foster that community, not to be *a perfectly pure Emacs user*. I use Jekyll and markdown for my website because I'm comfortable with the tooling and learning how to build an org-mode

website wasn't my goal. There's definitely nothing *wrong* with using org for your website, but don't lose track of your priorities.

Assuming everyone's a programmer

A bunch of scientists, mathematicians, and even fiction writers use Emacs. If *every* talk is about configuring Python, or whatever, you're gonna lose some people.

Trying to do it all yourself

You're probably terrible at a lot of things. Among them, most likely, are *design* and *video production*. Luckily, there are skilled professionals out there that do these things for a living; many of them are more experienced at these things than you are at, say, programming.

Unless you have some extensive design or video production experience, you probably want to outsource these jobs to professionals. You may have some friends with these skills who would be willing to help you out, or you might have to hire someone.

Either way, it's extremely worthwhile. EmacsNYC got an awfully nice logo designed by one of thoughtbot's designers (hi, Will!), which we turned into a sticker. All of our videos have been produced by Thom Obarsky, who also produces a bunch of thoughtbot's videos and podcasts. The stickers are mostly just for fun, but professionally produced videos are *terrific*.

The big takeaways

If you get nothing else out of this article, let it be:

- Get help: Find cofounders and sponsors, and use professionals to do things that you're bad at.
- Consistency is important. Meet regularly in the same space.
- Be *actively* inclusive.
- Don't get bogged down in technical minutiae.

Happy hacking!