

My org-mode workflow

✓

A [recent Hacker News discussion](#) inspired me to write a short post on my org-mode workflow, with the hope that it may help others.

The thing I like the most about org-mode is that it supports heavy customization. This is not surprising, since Emacs itself is all about customization. You don't just use an editor, you build the editor you want using this Text Editing Framework, so to speak.

I came to emacs precisely because of org-mode, after enough people recommended it. I came from vim and the first turn-down for me where the keyboard shortcuts. I partially improved that by using evil-mode, which in my view gives you the best of two worlds: the power of Emacs and the ergonomics of VIM. That is, of course, if you think VIM is an ergonomic text editor (I do).

The second huge improvement to the usability of Emacs (and org-mode) for me came via the Hyper key. I have written about this [before](#) and while my setup has changed a bit since that was written, the idea still holds.

I use a Kinesis Freestyle2 Blue keyboard and combined with heavy use of the Hyper key (all but 7 keys on my keyboard are currently mapped as hyper key shortcuts!) and slight typing changes (Never use the pinky for Shift, remap Caps-Lock as Control, that kind of thing), I can spend my typing time without incurring in physical pain.

So enough for the intro, here's how I use org-mode to manage my work and personal life:

Capture templates and a global Inbox

I use org-mode's [capture templates](#) to create new entries whenever I need to get something off my head but without losing track of it. Typically that means creating follow-up To Do items after checking my email, getting a new ticket assigned to me, etc. This lets me go through my backlog at the start of the day (I have a daily "check-in" routine for this) and triage every item in the queue to make sure I don't drop the ball on anything that's urgent, while still allowing me not to waste mental space trying to track all tasks.

When going through the backlog, I classify things in three categories:

- What I can do right now because I expect it to be fast. "Fast" is a subjective concept here. I never read the book, but from talking with others that have, my approach is inspired by [Getting Things Done](#), and I believe the author sets the threshold at something that takes 5 minutes or less.
- What I don't need to do right now but I must do at some specific time. Things with a deadline. These go to a ToDo entry that is scheduled and has a deadline (Going back to my comment about shortcuts, I have Hyper keys setup to do both these things). It's important to set this properly so that I can feel confident that once I'm done with the capture process, I can move on to the next item in the backlog and forget about this one, while still knowing I won't drop the ball on it.
- What I would like to do eventually, would be nice if I do, etc. Things that fall in this category tend to be ideas to improve a project, write a blog post, learn something new. Stuff I'd like to remember as "I would love to do this Some Day" but that is not time sensitive. I typically go to these items whenever I'm stuck, or I find myself procrastinating.

By default, my capture templates go to the global Inbox. This is a heading in my Todo.org file. I'll talk more about files in the next section but for now what matters is that I will usually refile items to their file, or to their header in the global one.

By the end of this process, I have a better picture of how my day (and upcoming days) will look like, and I can plan how to tackle it accordingly.

My org files

As mentioned, I have a Todo.org file, which I call my "global" file. It holds my Inbox, my Personal heading (where I file anything personal that doesn't merit a file of its own) and a Trashcan. I borrowed the idea of a Trashcan for ideas as opposed to just deleting the "bad ones" right away from Robert Pirsig's description of his workflow in [Lila](#). He used cards and my analogy is an org-mode heading (or subheading, etc) is the equivalent to one of his cards.

Besides this, I have some things in their own org file. Examples include:

- Community.org to track work on community contributions, including ideas of the “would be nice to do” type, but also work with (usually soft) deadlines on bug reports and blog posts.
- One file per each large-enough project I’m involved in at the moment, and/or for each active client. When going through my email, if I have a request from a client, after capturing it I will refile it to the appropriate file.
- On-demand ephemeral files for things that I need on the spot. Say I need to hop on a call on short notice. I used to add a headline to an existing file and work there. Now I just create a file only for this, work there, and if needed, refile whatever is important to another place when the call is done. Otherwise, it’s easy to move the file to the Attic and be done with it. One of the goals of my setup is that I can both minimize “waste”, which I define as things that I don’t need and clutter my attention, while still letting me keep track of where I’ve been (something important so you have a decent chance of getting to where you’re going). I refer to this exit-target as Attic instead of Trashcan because I typically don’t delete things, unless I’m done with a client and the contract requires that I remove all information about this. Otherwise, having a history of my work as back as I have been doing it is quite useful to plan new things. Org-mode is all about text files so with today’s drives, there’s no way you can ever have disk-space problems because you have too many org files.

My snippets

I make use of snippets (via [yasnippet](#)) to help me work faster and avoid errors in repetitive work, but also to keep track of my steps. Specifically, on this last item, I have a ‘tsnow’ snippet that expands to the current timestamp (date + time) and when working on a case (say, a bug report), I will create an org header with a brief description of what I’m doing and then invoke the snippet. This creates a worklog that I can go back to when needed and is immensely helpful to me, especially if I need to interrupt my work for some time and then go back to it. When this happens, I also create a special log header called “Status at <tsnow>” where I do a short brain dump of what I’ve found so far and what I think my next steps should be. A nice side effect is that I can export the worklog in plain text (or HTML or PDF too) and upload it to a ticketing system.

Within each timestamped header I also make use of snippets to create EXAMPLE and SRC blocks as needed, to include code snippets and console output of what I have been doing. For more info on these type of blocks, I refer you to [the fine manual](#).

Quite often, these worklogs become narratives with code interweaved. I like to think of this as “Literate Troubleshooting”, but that may be just because I am a heavy Knuth fan :)

Agenda

Org provides an agenda view and while by default it works fine, I have a custom view that shows me anything that is scheduled for today (or for another day in the week, in the weekly view) in the IN PROGRESS, TODO, and SCHEDULED states. In org mode, headers can have a state, and those are useful to [organize your workflow](#). Mine are: TODO, IN PROGRESS, SCHEDULED, DONE, WAITING FOR FEEDBACK, WAITING ON CUSTOMER, SUSPENDED, REASSIGNED, and CANCELLED.

Focusing

While org-mode is very nice to organize your thoughts, tasks, and many other things, multi-header org files can get cluttered quickly. If this becomes too much cognitive noise to you (it does to me at times) I recommend using the focus and widen options. I have each of this assigned to a Hyper key so that I can easily make anything except that heading I’m currently interested in, disappear, and then reappear as needed.

Tracking time

If tracking time is important for you, org’s [clock](#) will come in handy. I basically run with the clock on all the time, and just have it switch automatically as I go from heading to heading, which is kind of saying from task to task. Off for lunch? No problem, I just use my “Interruption” capture template before going afk. As a result, at the end of the week, I can get a nice report of where I’ve spent my time.

I hope you found this useful, and I hope you give Emacs and org-mode a chance. They’re both beautiful software that can help you get a lot done.

Written by

Fernando Ipar
