

A GUIDE TO MY ORGANIZATIONAL WORKFLOW: HOW TO STREAMLINE YOUR LIFE

SUN 22 MAR 2020
Gregory J Stein

TAGS
Meta Emacs

Five years ago, my life exploded in complexity. I had just started a new position in a new field. I was planning my wedding. And my inability to say **NO** to anyone and everyone had culminated in my serving on the board of *three* graduate student organizations. Inevitably, cracks began to form, and my finite brain started to lose track of tasks. My calendar was sufficient to ensure that I wouldn't miss meetings, but I would often only prepare for those meetings at the eleventh hour. My productivity and the quality of my work both suffered. Something needed to change.

This guide is devoted to a discussion of the organizational system that I have honed in the time since. With it, I have found that my time is spent more wisely. Better organization means that I can consciously devote effort where it is needed early on, as opposed to scrambling to keep up, and deliver higher quality work without expending more energy.

You too can streamline your process. This guide is meant to serve as an example of how you might reorganize your workflow and find order through the chaos of your busy life. Yet different lifestyles have different demands: what works for me may not work as well for you. As such, I do not expect that you will replicate this system in its entirety. Instead, I hope you will take inspiration from my system and use elements of it to build a workflow that works for you.

This document is broken into three main parts:

- **Goals:** in which I dive into more detail about what it is I have tried to accomplish with my system.
- **Framework:** in which I describe the core ideas and systems I employ to record information and keep track of my tasks.
- **Tooling:** in which I discuss the tools—including hardware, software, whatever—that I use to implement the framework.

In addition, I conclude with two sections in which I describe what I see as limitations of my existing system and some other technical details.

Let's dive in.

GOALS

In order to determine how effective an organizational system is, it is important to have clearly enumerated aims: what is the system being designed to enable? Your goals may be different, but my system is structured to prioritize the following:

- Ensure that I never miss a task, meeting, or deadline;
- Manage tasks that I share with and have assigned to others;
- Keep a permanent record of my work and research, and in a way that can be easily shared with others if necessary;

Many of the ideas presented here derive from the Getting Things Done methodology, but adapted and expanded to meet my personal needs.

- Collect my thoughts, writings, and half-baked ideas;
- Ensure that my local progress—on a daily or monthly basis—is in service of my long-term goals;
- Finally: ensure I enjoy my life and make time for fun and friends! *My system is meant to organize, not confine.*

A NOTE ON FLEXIBILITY AND ROBUSTNESS

One key idea to keep in mind as well is that an organizational system should be **flexible**. Life is complicated and unexpected items—both good and bad—can appear at a moment's notice. While it may not be immediately clear where a new task or project belongs, one should always have the ability to add new files or lists to which the new items can be added. Additionally, I do occasionally discover a file or project will outgrow the way I decided to structure it at its inception. My tools support fast and easy refactoring when necessary so that I can restructure a project to reflect my updated understanding of the problems it was intended to solve. The greater the effort required to reorganize when necessary, the less frequently it will happen and the effectiveness of the organizational system will decline. The tools I will describe later on work well for me, but you should find those tools that work best for you.

Similarly, an effective organizational system is **robust**: I am human and will occasionally forget to write something down or call someone back. The *Getting Things Done* methodology—upon which much of my workflow is based—recommends a weekly time of reflection during when I read through my outstanding tasks and rack my brain for anything I may have missed; I find this weekly effort an invaluable way of ensuring that my todo lists are complete. However you decide to organize your life, building in extra redundancies or time to tidy up is incredibly important.

Flexibility and robustness are properties that any organizational workflow should have if it is to be effective.

FRAMEWORK: GETTING THINGS DONE (WITH EXTENSIONS)

One of the core principles of the *Getting Things Done* methodology, around which my system is built, is that *the mind is for thinking, not remembering*. Everything that may need to be accomplished—or that you might someday want to accomplish—should be written down. This core idea is used to motivate expansive list-making. *Getting Things Done* (or GTD) is all about making comprehensive lists of things that need to get done. Things that have deadlines. Things that rely on other people. Things that have other sub-things before they can be completed. Lists of tasks give an at-a-glance overview of the big picture and can be extremely useful for accomplishing long-term goals.

Yet *Getting Things Done* treats notes and other information as if they are separate from the tasks that motivated their creation—but *this is not how most people think and work*. My usual workflow for solving a problem involves breaking up a high-level objective into increasingly smaller goals until I can make progress towards accomplishing it. As I work, I log my progress, my thoughts, where I get stuck, temporary images or figures, intermediate results, how to reproduce my work, and so on. Note taking is a critical part of my thinking process, so notes and their parent tasks should coexist.

Detailed notes are needed to accomplish individual projects or tasks but not the big picture. Task lists and calendars focus on the big

Contemplating how my day-to-day work aligns with my long-term goals is perhaps the most difficult objective, since long-term goals are difficult to even enumerate; I discuss how I think about and manage long-term goals in a later section.

Software—the subject of a later section—allows me to collect my tasks and projects when needed and build the lists around which *Getting Things Done* (and my workflow) is centered.

picture at the expense of detail. An effective organizational system requires both.

A ILLUSTRATIVE EXAMPLE: TAKING A CLASS

Imagine you're taking a class.

Succeeding in your class requires that you keep track of a plethora of information—including lecture times, class notes, assignments and their due dates, and term projects. Here's what an example file for this class might look like during the term:

```
* Class Lectures :NOTES:

** Lecture: Linear Regression
<2020-03-17 Tue 10:00-11:30>
This task has a date and time.
[Class notes will go here.]

** Lecture: Support Vector Machine
<2020-03-19 Thu 10:00-11:30>
[Class notes will go here.]

* Homework

** NEXT Homework Assignment 2
DEADLINE: <2020-03-17 Tue>

** NEXT Homework Assignment 3
DEADLINE: <2020-03-24 Tue>

* PROJ Mid-Term Project
DEADLINE: <2020-03-21 Fri>
This is marked as a project, since it has multiple
sub-tasks.

** NEXT Find four reference papers
We need to find four papers to write about. We can
make immediate progress towards this task, so it is
marked as NEXT (instead of TODO).

** TODO Choose one paper to implement
One of the papers we find, we will need to implement
for our project. This task requires that we have
completed the previous task, so it is marked as
TODO instead of NEXT.

** WAITING Check with Prof. paper topic okay
Sent email to Professor. Waiting for reply.

* Reading
** SOMEDAY Read ML Textbook :SOMEDAY:
When I find the time, I would like to read the course
textbook in more depth. The SOMEDAY tag means that
I hope to accomplish this task eventually or may
never make time for it.
```

The structure of the file is simple enough. Top-level items like *Assignments* and *Class Lectures* collect calendar items and standalone tasks as they are made clear. So where is the difficulty?

Well, if you're only taking this one class, managing your course load is pretty straightforward. But most students are taking multiple classes at once. They participate in social events and clubs. They have lives outside of the university. Rent and tuition need to get paid. The number of different organizations to which a typical student is somehow beholden can be overwhelming, and a separate log file may exist for each and every one of them. The mental burden of keeping everything straight—much less accomplishing anything—introduces massive cognitive load.

That's where the difficulty lies.

My system uses the Org syntax, similar to markdown, to store information. Asterisks are used to indicate the *level* of a task, so a task marked by ****** *belongs* to another task heading with *****.

The second part of my system involves building calendars and todo lists that aggregate tasks from across these files—usually automating the process in software. A Weekly Calendar view might look something like this:

```
Monday      16 March 2020
Tuesday     17 March 2020
  ML Class:   10:00-11:30 Lecture: Linear Regression      :NOTES:
  Phys Class: 12:00-13:30 Lecture: Incompressible Fluids :NOTES:
  ML Class:    DEADLINE: NEXT Homework Assignment 2
Wednesday   18 March 2020
  Dance Team: 19:00-22:00 Practice
Thursday    19 March 2020
  ML Class:   10:00-11:30 Lecture: Support Vector Machine :NOTES:
  Phys Class: 12:00-13:30 Lecture: Laminar Flow           :NOTES:
Friday      20 March 2020
  Phys Class:    DEADLINE: NEXT Homework Assignment 2
  ML Class:    DEADLINE: PROJ Mid-Term Project
Saturday    21 March 2020
  Dance Team: 12:00--18:00 Competition
Sunday      22 March 2020
```

Notice that the calendar collects tasks, meetings and deadlines from across multiple areas of focus. At a glance, you can see what the week is going to look like and what upcoming deadlines may require your immediate focus. The list of NEXT Tasks may look something like this:

```
Dance Team: NEXT Download video of new routine
Dance Team: NEXT Register for competition
Hobby:      NEXT Sign up for guitar lessons
ML Class:   NEXT Homework Assignment 2
ML Class:   NEXT Homework Assignment 3
Phys Class: NEXT Homework Assignment 2
Phys Class: NEXT Homework Assignment 3
```

These task lists are useful for gaining perspective and focusing effort where it is needed. Reviewing various task lists as part of a weekly agenda review is an effective way to ensure that tasks are not missed and that progress is being made towards longer-term goals. The remainder of this section will discuss each of these pieces in more detail.

OVERALL STRUCTURE: LOGS AND LISTS

In my system, everything is a type of *task*. Every task corresponds to something I should do, or somewhere I should be, or something I should remember. As such, a task can have many different types of data:

- *Tasks have notes.* Every task can have any kind of notes associated with them. Simple tasks, like chores or errands, typically have no notes. More complex tasks—especially items associated with my job—can have extensive logs that include external links, images, or executable code snippets.
- *Tasks have state.* Keywords like TODO, DONE, and CANCELED are used to indicate how complete a task is. WAITING is used to indicate tasks that are waiting on someone else's action before I can make progress towards them. Items can also be created without any state at all, and are useful for aggregating information or tasks of a particular type, like *chores*.
- *Tasks can have times and deadlines.* Adding time information to a task should be simple. Tasks associated with meetings or events have a time or time range associated with them. Deadlines can be added as well: times by which a task is expected to be marked as either DONE or CANCELED. Additionally, tasks can have a *defer date* until which they won't appear on any list of incomplete tasks.

Strictly speaking, my system also has support for *reference material*: notes that do not pertain to specific task, like course notes and the like.

- *Projects are just tasks with subtasks.* If I feel a task is too complicated to complete without further decomposition, I may create subtasks within that task. All subtasks need to be completed before a project can be marked as DONE. *Note:* I usually use the PROJ state to indicate projects for easier searching when building lists.
- *Tasks can have additional metadata.* This might include *where* the task will occur (useful for meetings) or *tags* that I can use for sorting and filtering of information. I have included a more complete discussion of how I use *tags* in the appendix.

My organizational system consists of collections of tasks with various amount of metadata associated with them. As I get new tasks and projects I write them down. Making progress towards projects or higher-level objectives often reveals new tasks. I usually have a separate file for different organizations or areas of focus, and each file is populated with its own projects or tasks. When I've decided to work towards a particular project or area of focus, I'll open that file and get to work.

But how do I know what needs to get done? For this, the *task lists* are very important.

My standard *agenda view* shows the **Weekly Calendar** and the **NEXT Tasks** list. When I start my day, I look at upcoming meetings and deadlines. I usually prioritize those, since they are time-sensitive. With those out of the way, I will look to all NEXT tasks to find tasks that I feel like doing. I will often use *tags* to filter my tasks by context: when I'm at work, I will only select tasks that are relevant for my job. My goals and deadlines are usually sparse: I have bigger projects that I need to deliver on, but whose deadlines are often months apart. Every day, I will usually pick one or two projects I want to make progress towards solving and start working on it. Projects are made up of smaller tasks: to make progress towards completing it, I can pick one of those tasks and start working on it. I continue to work on tasks or projects until the day is done.

Ranking tasks by *urgency* is generally more trouble than it is worth. My list of NEXT tasks is usually long but rarely is it terribly overwhelming. Filtering by *tags* is also occasionally useful.

THE WEEKLY REVIEW

On a day-to-day basis, new tasks are easy to keep track of. I write down tasks and projects as I become aware of them. Emails, conversations I have with friends or colleagues, random thoughts or blog posts—all of these can become new items to add to my logs. Even movies I want to watch or restaurants I want to go to can be added as SOMEDAY items. But reactively capturing tasks is only part of a well-organized existence. To facilitate progress towards goals and to encourage long-term personal growth, periodic reflection is required.

Consistent with the *Getting Things Done* methodology, every week I conduct a complete **Weekly Review** of my logs. The GTD weekly review consists of three phases:

1. Get Clear First, I ensure that everything not in my logs is added to my logs. Loose scraps of paper and notes are collected, processed, and discarded. My email inbox is emptied. Open browser tabs are reviewed and closed if possible. If I can clean it up and put it in its place, I do so.

2. Get Current Second, I look over *all of my tasks* and try to identify places where my logs might be incomplete or out-of-date. I look at all my NEXT tasks and remove any that have been completed. I review my WAITING tasks and follow up on any that may be overdue. Projects with no NEXT items are considered *stuck*; I update these if possible or flag them for later review. I reexamine tasks that have gone untouched for a long time: will I *ever* do these? I even review my calendar. Items from the last two weeks may need additional followup tasks, and deadlines or meeting for the upcoming week may require auxiliary tasks for preparation.

3. Get Creative Finally, I look beyond my existing projects and tasks for new opportunities. I review my list of long-term

The *Get Current* phase is designed to make the system more robust. It's hard for items to become stale if I look them over at least once a week.

goals—more on *goals* and *motivators* in the next section—in search of new tasks or projects. Can repetitive tasks be automated? Do certain aspects of my job feel overly stressful? Would gaining a new hobby or skill make me happier or more productive? Do I relax enough? I take a look at items tagged with SOMEDAY for TV shows, movies, games, or restaurants I've wanted to enjoy. In dedicating time to introspection, I can identify areas in which I feel I have not been making much progress and try to find ways my life might be improved.

The Weekly Review is essential for keeping my system up-to-date. Yet more important is ensuring my projects and tasks align with my long-term goals: if my system isn't making me happier or more fulfilled, why bother?

I have included an *exhaustively detailed* discussion of my personal weekly review in the appendix.

FROM SHORT-TERM TASKS TO LONG-TERM PROGRESS

When I started organizing my life, long-term goals were not on my mind. I was interested in little more than keeping my head above water—in finding some semblance of order through the chaos. Yet as I started to become more organized and think more than a week into the future, thinking about long-term goals helped to put my life into context. Without context, my accomplishments started to feel short-lived. Asking deep questions about motivation can help set direction, focusing time where it will be more useful in the long term. What sort of work did I want to do after graduation? How could I better spend my time during my PhD to get the job *I wanted*?

A key part of my organization system is my **List of Motivators**. Anything that might potentially generate new tasks or projects belongs on this list. In my system, motivators fall into three broad categories:

1. **Active Motivators** Any regular commitments, organizations I'm a part of, or ongoing activity is an *active motivator*. As a PhD student, I was a part of a number of student organizations; dedicating thought to each during the weekly review, I would occasionally come up with new events or try to identify ways in which those organizations might improve. This blog is also one of my *active motivators*—I try to think of new content or ways of improving the quality or visibility of my online presence. Hobbies can also fall in this category and my *learn guitar* motivator often inspires new songs to learn or techniques I would like to try.

In *Getting Things Done* parlance, what I call *active motivators* are referred to as *areas of focus*.

2. **Tangible Motivators** Sometimes a project is too small in scope to capture a milestone you would like to achieve. For me, graduating with my PhD was an example of this. Graduation required (1) writing my thesis, (2) passing my *defense*, an hour-long oral presentation, (3) satisfying course requirements, and (4) publishing the research upon which much of the thesis was built. *Tangible Motivators* are typically milestones that are greater than a few months into the future, yet might require indeterminate intermediate steps to accomplish. Writing a book or getting a promotion are other good examples of *tangible motivators*.

3. **Conceptual Motivators** The broadest long-term goals—like creating a successful business or becoming a thought leader in my field—belong to this class of motivator. I often consider my *tangible motivators* in the context of their *conceptual* counterparts. My goal to graduate from my PhD was connected to my desire to become a professor and to create an influential body of research in my chosen discipline.

The weekly review has a dedicated time during which I think critically about my different motivators. I identify ways I might make progress towards them, adding new tasks or projects to my logs in the process. Reviewing this list allows me to think critically about when my goals are not being met. Yet finding new tasks is only part of the process of reflection. I need to decide if my motivators themselves reflect what I want. Are there any missing? Do I no longer care about any? Critically, are there tasks I have enjoyed working on that don't have a corresponding motivator? The weekly review should not feel utilitarian: if I find myself with a handful of projects that lack a corresponding motivator, perhaps something is missing.

Some soul-searching is obviously required here, since I cannot

tell you what your goals are or how you should make progress towards them. Habit is the key: *regular* self-reflection has helped me find direction.

TOOLING: EMACS AND ORG-MODE

Understanding how my organizational workflow is *meant* to work is only one part of understanding *how* it works. The workhorses of my system are the tools—the services, software, and hardware that I use to get everything working.

PROJECT MANAGEMENT WITH ORG MODE

My primary tool is Org mode, a notetaking syntax and surrounding software environment built-in to the customizable text editor Emacs. To be clear: *Org mode and Emacs are not tools for everyone*. Here is an overview of the highlights:

Logs are written in plain text with simple markup. The *class example* from earlier shows a minimal Org mode file. *'s at the beginning of a line are used to indicate new headings, projects, or tasks. Other markup is used to stylize text: ***bold text*** becomes **bold text** and */italics/* becomes *italics*.

Keywords are used to indicate state. Task states like NEXT or SOMEDAY are identified automatically by their context, as in the *class example* above. A single function, org-todo, is used to quickly change the state of a task. During the weekly review, the keywords are needed for creating the task lists.

Moving tasks around is fast. Sometimes I want to move tasks to another location, particularly during the weekly review. With Org mode, this is done with a simple function: org-refile.

Capturing new tasks is simple. Many new tasks come in as emails or during meetings. For this, org-capture is incredibly useful. With it, I can create a new task with a single function call from one of a number of custom templates and then move it to where I would like it to go using org-refile.

Searching is rich and powerful. Since my Org mode files are just plain text, searching through them is straightforward. Org has a couple of extensions that allow me to search across all of my log files. I can even search by date, title, file, content, tags, metadata, anything with relative ease.

Clocking my time is simple. Clocking my time is not essential to my workflow. However, I find that it is a great way to keep track of how much effort I put towards different tasks and reevaluate if I'm appropriately allocating my time to one project over another. During my weekly agenda review, looking back over the last couple weeks of clocked time is also useful for reminding me what I have been working on recently. I usually only use this for work items—it doesn't really matter to me how much time I spend on my hobbies or playing video games—but I do still record how long it takes me to write blog posts or other professional development activities. Clocking my time with Org mode is trivial and requires only a simple key sequence to enable or disable the clock for a given task.

Inline citations and references are easy with org-ref. It is very important to me that I can include references to papers and other publications in my notes. Since I work in academic computer science, I frequently use LaTeX for composing documents and BibTeX for managing references. With org-ref, written by the fantastic John Kitchin, I can add citations from my BibTeX file in my org-mode notes with a single command. Whenever I export my notes to a PDF via LaTeX, the citations appear inline and optionally populate a bibliography auto-generated at the end of the file.

Exporting and sharing my notes is simple and flexible. As you might have guessed, I do nearly everything in Org mode. Sharing notes with my colleagues is a critical part of my workflow and Org mode makes that easy with its flexible support for exporting and publishing notes. Turning a project into a PDF, via LaTeX, or a stylized web page is never more than a few commands away. I even have a custom exporter for my blog so that I can automatically insert Tufte-style the figure captions and margin notes I love to use.

I have a slightly more technical writeup about my Org mode and Org Agenda workflow in another blog post as part of my Emacs for Productivity series.

Org mode even supports inline code snippets that can be run without leaving the file. I wrote about this feature in a post on Literate Programming with Org Mode.

GENERATING TASK LISTS WITH ORG AGENDA

Org mode is what provides the core functionality of my logs, but Org Agenda is what makes it all work. The agenda is *the killer feature of Org mode*. It is through Org Agenda views that I can build my calendar directly from my logs and create the task lists needed for daily operation—via the `NEXT` Tasks list and the `Projects` list—and my weekly review. I have custom task list generators for nearly all elements of the *Get Current* phase of my weekly review. In addition to simply aggregating tasks, the agenda view can automatically determine when a `PROJ` is *stuck* (does not have any `NEXT` tasks) or is *waiting* (is blocked by a task that is in the `WAITING` state). The views the system produces mirror those from the *class example* above.

Agenda views themselves are not merely plain text outputs: they are interactive. Each task is a link that, when opened, brings me to its corresponding location in my Org mode logs. The agenda can be filtered as well and I can easily narrow down what items I'm shown by their name, tag, or other metadata—this is useful when I'm at work, and I probably shouldn't be working on my hobby projects.

Note: My workflow for managing tasks assigned to others or collaborative projects is still a work in progress. The system is tentatively working as expected, but I am not comfortable writing it up just yet. For an idea of some of the functionality I use, you can get an idea of how it works from [this guide to team management with Org](#).

Note: My more in-depth guide on creating custom agenda views and task lists is still a work in progress. For now, you can look at [my technical overview](#) or the [incredibly detailed guide by Bernt Hansen](#).

OTHER TOOLS

Org and the Org Agenda are incredibly powerful, but general-purpose software isn't always the easiest to use. Sometimes other tools are more appropriate.

My smartphone I can't always be near my laptop to add new tasks. Fortunately, I have another tiny computer in my pocket that allows me to write down short notes that can be added to my logs at a later time. I usually just use the Notes application on my iPhone. The `beorg` iOS application is very easy to use and syncs with my Org files over Dropbox, allowing me to perform more complex operations on the go if necessary.

My email My relationship with email is constantly evolving. For a long time, I used the built-in desktop Mail client on my MacBook Pro. There was a time when I used Emacs to manage my email, something I wrote about extensively in [another blog post](#). Now, I mostly use my desktop mail client again, though occasionally use my Emacs-based interface to sort through my mail and even link tasks with their associated mail message.

Staying in touch with friends and colleagues using "Monica" Though I once managed all my contacts in Emacs and Org mode, I have since found it easier to have a dedicated and specialized tool for this purpose. `Monica` has been my go-to tool for this purpose. It's open source and optionally web-hosted for a small monthly fee. `Monica` is advertised as *personal Customer Relationship Management (CRM) software* and, with a clean web interface and flexible information storage, it delivers.

Personal finances with Ledger CLI My finances are not terribly complicated. I use `Ledger`, a command-line-based utility for double-entry accounting, to manage my money and budgets. Like Org mode, `Ledger` has a pretty simple syntax for storing entries. An example file looks something like this

2019/11/01 iTunes	
Expenses:Media:Music	\$9.99
Liabilities:MasterCard	
2019/11/01 Safeway	
Expenses:Food:Groceries	\$35.67
Liabilities:MasterCard	
2019/11/02 Compass Coffee	
;; Coffee and lunch with my wife	
Expenses:Food:Coffee	\$8.00


```
Expenses:Food:Snack          $7.88
Liabilities:MasterCard
```

Computing the balance of each account is simple:

```
gjstein $ ledger -f example.ledger balance
$61.54  Expenses
$51.55   Food
$8.00    Coffee
$35.67   Groceries
$7.88    Snack
$9.99    Media:Music
$-61.54  Liabilities:MasterCard
=====
0
```

So too is generating the history for a particular account, like Food:

```
gjstein $ ledger -f example.ledger register Food
> 19-Nov-01 Safeway      Ex:Foo:Groceries  $35.67  $35.67
> 19-Nov-02 Compass Coffee Ex:Food:Coffee   $8.00   $43.67
>                      Expen:Food:Snack  $7.88   $51.55
```

Ledger is a simple yet powerful tool that I strongly recommend.

LIMITATIONS

As with any system, there exist shortcomings. There aren't many –certainly not enough for me to consider another major overhaul –but what shortcomings I've identified I include here.

- *Archiving*: I do not have a very principled way of deciding when material should be archived, yet if material is never archived, clutter can quickly become overwhelming.
- *Reference material*: It is hard for me to know where new knowledge should be kept. One of my log files is for an expansive annotated bibliography, separated by category. As a researcher in an active field, many new publications blur the lines between subfields, making it challenging to know where to put that information. More specialized programs like Papers or Mendeley seem great at managing this sort of thing, but I haven't had the impetus to leave the comfort of my system just yet.
- *Laziness*: I am still human, and I occasionally fail to perform the weekly review. For one week, maybe two, this isn't a problem. Over longer periods, cracks form. I can no longer trust that my system is a complete list of my tasks, and that lack of confidence makes the system significantly less effective.

CONCLUSION

I hope to have inspired you. My organizational workflow transformed my life, empowering me with the ability to plan for the future, rather than reactively making progress towards short-term goals. You should now have all the ideas you need to put in place a system of your own. Take control of your life and find order through the chaos.

Good luck.

As always, I welcome discussion in the comments below or on Hacker News. I encourage you to ask questions, provide feedback, or let me know if you might like more detail in a follow-up guide on a particular area.

APPENDIX: SUPPLEMENTARY DETAILS

A COMPLETE LIST OF TASK STATES

Without any customization, Org allows you to assign a TODO or a DONE states to any heading, either manually or using C-c C-t. However, I've found that these aren't typically expressive enough for handling very large projects. I use the following keywords (and shortcut keys) for my documents:

- NEXT (n) This is for tasks that can be immediately worked on.
- TODO (t) Used to indicate blocked tasks, which require another task to get done before I can start working on it.
- PROJ (p) Denotes a project, which usually contains other TODO or NEXT tasks. Projects without any NEXT tasks are considered stuck.
- WAITING (w) Whenever I have a task that is waiting on someone else, I'll assign it this keyword.
- INACTIVE (i) As an academic, I will occasionally have ideas or projects that I'll want to get done eventually, but won't have time to work on them at the moment. I typically use the INACTIVE state to signify that I'd like to come back to it eventually and I'll occasionally search my files for INACTIVE projects when I have time or during my weekly review.
- CANCELED (c) This is self-explanatory. If I'm working on something for a while, but it fizzles, it's sometimes useful to mark it as canceled rather than delete it outright.
- DONE (d) This should be self-explanatory.

When changing between these states using C-c C-t, it's possible to automatically prompt the user for clarification on what prompted the change. For example, whenever a task is set to WAITING, I have an opportunity to enter a short message detailing who I'm currently waiting on, and any additional information I may want to add. In addition, this command may also automatically add tags (e.g. CANCELED) to the heading to help with sorting and searching.

MY WEEKLY REVIEW IN EXHAUSTIVE DETAIL

Much of this section is inspired by, or derives directly from, the principles outlined in *Getting Things Done*. However there are a number of things I've added or expanded upon to really make it my own. Your weekly review likely won't match mine exactly, but hopefully it will give you an idea of what an exhaustive review requires.

Get Clear Everything that I want to be in my logs should be added to the logs.

- [] *Clean out my email inbox*: I habitually use my email inbox to store messages that I should probably respond to. Either respond to it now, add a task to respond later, or forget about it. Regardless, it should be moved out of my inbox if possible.
- [] *Prune browser tabs*: If you're like me, you probably have dozens (hundreds?) of browser tabs open at a time. Many of these are papers or blog posts I may never read. I figure out the ones I want to read and write them down. The rest are closed.

Sometimes, it may not be clear

- [] *Collect loose papers and materials*: Any handwritten notes or items I've added to my iPhone's Notes app should be transferred into the system.
- [] *Empty your head*: Anything that's been on my mind should also be added. Usually, this step doesn't mean much, but occasionally I'll remember something I had to get done or someone I wanted to follow up with.
- [] *Clean my refile list*: Any new tasks that I don't know where to put during the week sometimes end up in a temporary refile list. Here, I look through that list and move tasks to where they belong, creating new headings, files, or projects if necessary.

Get Current Every open item in my logs—incomplete tasks or projects—should be looked at.

- [] *Review my daily log from the last week*: I like to keep a pretty simple log of things I accomplished at the end of any given day, including what went well or where I got stuck. Looking over that log helps me come up with tasks I may not have thought of at the time.
- [] *Review the last 2 weeks of calendar entries*: Similar to the daily log review, this occasionally reminds me of follow-up tasks or items that I may have missed.
- [] *Review the next week of calendar entries*: Any upcoming deadlines may need additional tasks.
- [] *Review the NEXT Tasks list*: Sometimes I forget to mark a task as complete; looking through all my NEXT items prevents the system from getting stale.
- [] *Review the WAITING Tasks list*: The weekly review is a good time to make sure I reach out to people who may need prodding to get something done.
- [] *Review my list of open projects*: I should spend a couple of minutes looking over every project, marked with a PROJ task.

Get Creative Finally, I look for opportunities for personal growth. The Getting Things Done book has some great suggestions for how to think about finding new projects or motivators. I highly recommend picking up a copy—and reading Chapter 7 in particular.

- [] *Review my inactive and "someday" lists*: First, I take a look at projects that I've put on the backburner. The SOMEDAY state is devoted to projects, tasks, and other items associated with *wishful thinking*. I move any SOMEDAY or INACTIVE items to a more active state or cancel them as I see fit.
- [] *Review my motivators*: This task is by far the most difficult of my review, which is why I have saved it for last. I go over my *Motivators List* and ask myself if I'm making sufficient progress towards them. What might I do to improve them? Are there any motivators that I'm no longer excited about? Are any motivators missing?

where a new item belongs. For this, I have a **Refile List**, a specific heading with the **REFILE** tag where I put tasks for organization at a later date.

TAGS

Many of my projects and tasks have metadata associated with them in the form of *Tags*. Tags are not *essential* to my

workflow, but they are a useful way of adding some additional info to tasks that helps with filtering and sorting. Here are some of my commonly-used tags and use-cases:

- I often append the **RESEARCH** tag to projects relevant for work. During my weekly review, I can use such tags to focus only on one set of tasks at a time, reducing context switching. Tags like **BLOG** or **HOBBY** serve a similar purpose.
- Some tags are location-specific; the **@HOME** tag is used to denote activities that I need to be at home to accomplish (many chores have this tag).
- I use the tags **TENT** (tentative) and **OPT** (optional) to denote meetings or events that I may not have to attend. My agenda view is set up so that any tasks or events with these tags are grayed out so that I can recognize them at a glance.
- I have a **INFO** tag that indicates when a task needs additional information. There are two reasons I use this tag: (1) the task is just a stub and needs some details, including metadata like due dates or location; (2) the task is completed, but I have yet to add any details. This is particularly important for my research, for which ensuring that I have a complete record of my tests and experimental results is essential.
- Finally, I occasionally use a **DOC** tag to denote *documentation* or other reference material. The tag both highlights information that I might want to put at the top of a project for reference and that the material may be subject to change, since documentation should be updated if procedures or other information updates.

Liked this post? Subscribe to our [RSS feed](#) or add your email to our newsletter:



+ SHOW COMMENTS FROM DISQUS