Org-Mode Workflow: Task Management

🕐 2020-10-31

As mentioned in the **last post**, I switched to Org-Mode. I kept adjusting my workflow with this new tool and it has been stabilized for a month. I think it is time to talk about the new workflow for task/time management with Org-Mode. This blog post consists of four parts: the principles, the definitions, the workflows, and finally the implementations.

1 The Principles

Principles remain valid no matter what the tool is.

1.1 Do Not Add Tasks Indiscriminately

Not every task should go into the system. Avoid filling the system with bullshits and hiding the things that matter. I only add tasks that I really want or need to do.

To clarify[1], the task management system described below is not the "inbox" in GTD. I still capture things into my inbox but not all of them will be converted to a task in the task management system (org agenda files) eventually.

1.2 Not All Tasks Have To Be Done

There are two reasons for this. First, tasks could be deprioritized or even become unnecessary. Second, we have limited time and cannot do everything. We should have an opinion on the priority.

1.3 Reduce The Number Of Open Loop

Open loops are tasks that have been started but not finished. They stay in our minds and occupy some of our limited working memory so that we cannot focus on another task we are working on.

Also, open loops reduce agility, according to **Little's Law**. The more the open loops, the longer time finish each of them on average.

1.4 Reduce Decision Making Of What To Do Next

The system should suggest to the user what to do next so that the user can reserve the will power to the real task. This also avoids skipping hard tasks with easy tasks unconsciously.

2 The Definitions

Each task in Org-Mode has a **TODO keyword**, optionally **a scheduled date, and a deadline**. For example,

```
* PROG Write a blog post on task management with Org-Mode
DEADLINE: <2020-11-07 Sat> SCHEDULED: <2020-10-31 Sat>
```

Each Org-Mode user could define their own set of TODO keywords and use scheduled dates and deadlines differently. For example, some people use only two TODO keywords, "TODO" and

"DONE", while some use more. Some people set "scheduled dates" to all the tasks while some people set it to some of the tasks. These nuances could result in a very different workflow, although they are using the same Org-Mode. Let's take a look at how I use them.
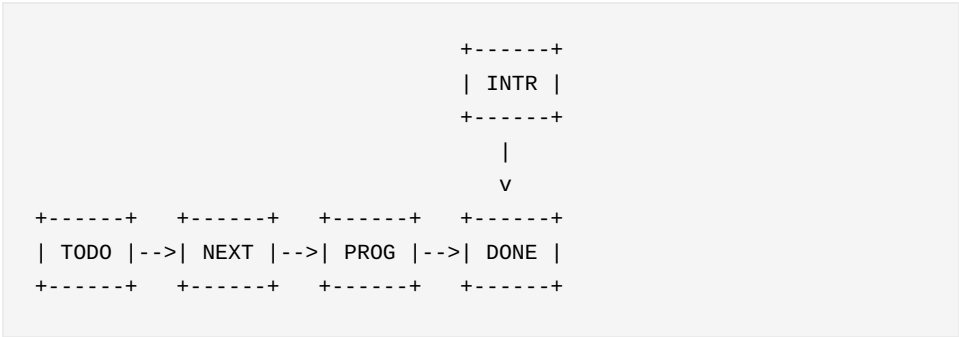
2.1 TODO Keywords

I use as few TODO keywords as possible but not too few. For example, it is common to use only two states ("TODO" and "DONE") but this does not align with the principles I mentioned above. I need a state for "open loops" so that I can keep the number of them small. I also need to distinguish a smaller set of "next actions" from all tasks.

So far, I defined these five keywords:

TODO Keyword	Definition
TODO	**Tasks that are not started and not planned.** They could be the backlogs or the GTD's someday/maybe. These tasks could be converted to NEXT during a review.
NEXT	**Tasks that are not started but planned to do as soon as I can.** When there is no actionable PROG (e.g., blocked), I start one of those and convert it to PROG .
PROG	**Tasks that are working in progress (open loops).** I work on these tasks before starting another NEXT task to avoid too many open loops at any moment.
INTR	**The tasks that are interruptions.** They are urgent things that I should drop everything else and work on it. For example, production issues.
DONE	**The tasks that are completed.**

This diagram illustrates the transition of those states.

```
                              +------+
                              | INTR |
                              +------+
                                 |
                                 v
+------+    +------+    +------+    +------+
| TODO |-->| NEXT |-->| PROG |-->| DONE |
+------+    +------+    +------+    +------+
```

2.2 Scheduled and Deadline

In the past, I tended to set a date for all tasks. If I want to do A, B, and C on Monday, then I schedule them for Monday. This sounds very intuitive but, in reality, I ended up rescheduling many incompleted tasks at the end of every day. It was not only wasting time but also depressing.

Later, I changed to rely more on the TODO keywords. For example, if a task is still in progress, I keep the state unchanged as PROG instead of rescheduling it every day until it is done. I am now using the "scheduled date" to hide a task until the date I should look at it again. Similar to the snooze feature in Gmail.
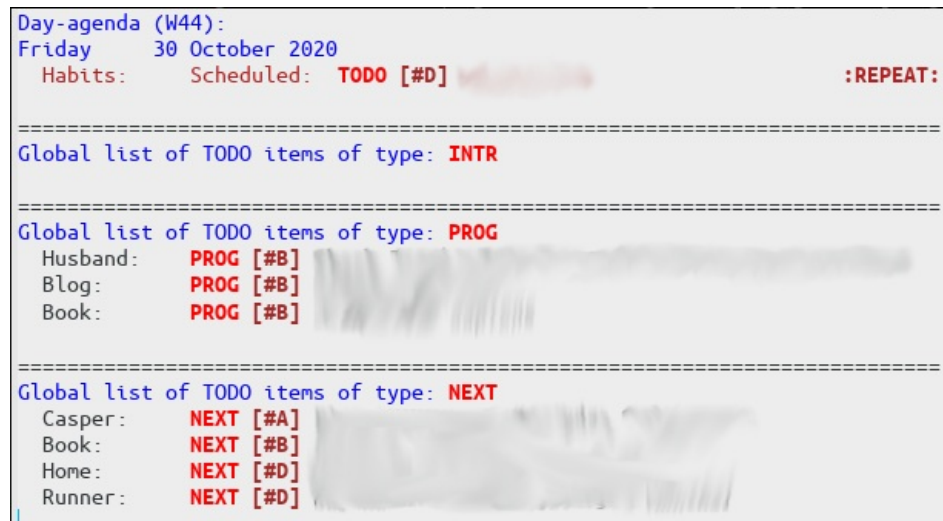
Date	Definition
SCHEDULED	Hide the task until the scheduled date.
DEADLINE	The deadline of the task.

For example, when a PROG task is being blocked, I set the SCHEDULED date to hide it until the date I want to revisit. On the scheduled date, if the task is unblocked, I will remove the SCHEDULED date. If the task is still blocked, I reschedule it again. It acts as the **waiting for list**

in GTD.

3 The Workflow

I customize my org agenda view to drive my daily workflow. The customized agenda view has four sections. From the top to bottom, they are the tasks scheduled today, the `INTR` tasks, the `PROG` tasks, and finally the `NEXT` tasks.

```
Day-agenda (W44):
Friday     30 October 2020
  Habits:     Scheduled:  TODO [#D]                              :REPEAT:

================================================================
Global list of TODO items of type: INTR

================================================================
Global list of TODO items of type: PROG
  Husband:     PROG [#B]
  Blog:        PROG [#B]
  Book:        PROG [#B]

================================================================
Global list of TODO items of type: NEXT
  Casper:      NEXT [#A]
  Book:        NEXT [#B]
  Home:        NEXT [#D]
  Runner:      NEXT [#D]
```

My daily workflow goes from the top to the bottom.

3.1 Update Tasks Scheduled Today

At the beginning of the day, I review the tasks that are scheduled for today. The goal here is not to finish them, but to update or remove the scheduled date so that there is nothing left.

1. If the task is still blocked, reschedule it
2. If the task could be done in a few minutes, then do it and mark it as `DONE`.
3. Otherwise, remove the scheduled date and optionally update the TODO keywords.

Removing the scheduled date is the best outcome. It indicates the previous estimation was correct, at least not too early. Rescheduling indicates the previous estimation is inaccurate. I would avoid rescheduling the task to tomorrow indiscriminately and try to make a good estimation to reduce the number of rescheduling.

3.2 Find the Next Task to Work On

After reviewing all tasks scheduled for today, it is time to pick a task and do some real works. This step is very straight-forward with the customized agenda view above.

1. Pick an `INTR` task if there is any.
2. If there is no `INTR` task, then pick a `PROG` task and work on it. If that task is blocked, set a `SCHEDULED` date to hide it.
3. If there is no `INTR` and `PROG` task, then start a `NEXT` task.
4. If there is no task in the agenda view, then review the `TODO` tasks and convert some to `NEXT`.

3.3 Review the System

The secret of having a system that works in the long-term is regular maintenance. I do it at least once a week. For examples,

- Promote some tasks from `TODO` to `NEXT`. Demote or even delete deprioritized tasks.
- Review the **journal** and add `TODO` if something needs follow-up.
- **Archive** completed tasks and extract to permanent notes[2].

4 The Configuration

Finally, here is the configuration for the above workflow.

```
;; TODO keywords.
'(org-todo-keywords
  '((sequence "TODO(t)" "NEXT(n)" "PROG(p)" "INTR(i)" "DONE(d)")))

;; Show the daily agenda by default.
'(org-agenda-span 'day)

;; Hide tasks that are scheduled in the future.
'(org-agenda-todo-ignore-scheduled 'future)

;; Hide the deadline prewarning prior to scheduled date.
'(org-agenda-skip-deadline-prewarning-if-scheduled 'pre-scheduled)

;; Customized view for the daily workflow.
'(org-agenda-custom-commands
  '(("n" "Agenda / INTR / PROG / NEXT"
     ((agenda "" nil)
      (todo "INTR" nil)
      (todo "PROG" nil)
      (todo "NEXT" nil))
     nil)))
```

1. Thanks for **this comment** in Reddit. ↵

2. There will be another post for Org-Mode note-taking workflow. ↵

🏷 **EMACS**   **PERSONAL-DEVELOPMENT**